BUSINESS LOGISTICS MANAGEMENT FOR CAMPESINOS USING BLOCKCHAIN

A PROJECT REPORT

Submitted by

SOWMIYA.R	**(211419104262)**
P.SREE LIKITHAA	**(211419104264)**
SHARON SONA.D	**(211419104249)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution,Affiliated to Anna University,Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **" BUSINESS LOGISTICS MANAGEMENT FOR CAMPESINOS USING BLOCKCHAIN"** is the bonafide work of **"**SOWMIYA.R(211419104262),P.SREELIKITHAA(211419104264), SHARON SONA.D(211419104249)" who carried out the project work under mysupervision.

SIGNATURE	**SIGNATURE**
Dr.L.JABASHEELA,M.E.,Ph.D.,	**MRS.V.SATHYAPREIYA,M.E.,(Ph.D.)**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
DEPARTMENT OF CSE,	**DEPARTMENT OF CSE,**
PANIMALAR ENGINEERING	**PANIMALAR**
ENGINEERINGCOLLEGE,	**COLLEGE,**
POONAMALLEE,	**POONAMALLEE,**
CHENNAI-600 123	**CHENNAI-600 123.**

Certified that the above candidate(s) was/were examined in the Anna University Project Viva-Voce Examination held on.................

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We "**SOWMIYA.R (211419104262) , P. SREE LIKITHAA (211419104264), SHARON SONA.D (211419104249)"** hereby declare that this project report titled "**BUSINESS LOGISTICS MANAGEMENT FOR CAMPESINOS USING BLOCKCHAIN**", under the guidance of **MRS. V.SATHIYA PREIYA, MCA., M.Phil., M.E., ASSOCIATE PROFESSOR** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors

Tmt.C.VIJAYARAJESWARI, **Dr.C.SAKTHIKUMAR,M.E.,Ph.D.**and **Tmt.**

SARANYASREE SAKTHIKUMAR B.E., M.B.A., for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** for histimely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. L.JABASHEELA, M.E., Ph.D.,**

for the support extended throughout the project.

We would like to thank our Project Guide **MRS.V.SATHYAPREIYA, MCA., M.Phil., M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

SOWMIYA.R
P.SREE LIKITHAA
SHARON SONA.D

ABSTRACT

The Brought together Nations 2030 conservative improvement goals hope to influence world economies toward a more commonsense future. These legitimacy targets mean watching out for natural concerns, decreasing Lop-sidedness, and keeping an eye on dejection, especially for the most vulnerable and troubled in the public field in this progressing monetary perspective. Current present day store ties add to differences and natural loads. These stock chains may similarly hold ensure for watching out for social and ecological ills. Uneven characters and ineffectually earth fragile pieces of the creation network appear in the most significant upstream bits of the store organization. For example, the Africa cultivation item store organization — beginning with its smallholder farmers — are vulnerable before various strong performers from spread out overall stock chains. In this perspective article, we depict issues defying these smallholder farmers and how development can expect a section for themselves along with their reserve chains to moderate different social and environmental ills. A few surprising models give judicious encounters. This discussion sets the supporting of critical investigation requests for advancement, improvement, and planning organization experts, focusing in development for social.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

1.1.1 The Java Framework

Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems& Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers & write once, run anywhere .Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications .The java framework is a new platform independent that simplifies application development internet. Java technologies versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to data centers, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

1.1.2 Blockchain

A blockchain is essentially a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. And, once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made. Bitcoin, the decentralized peer to peer digital currency, is the most popular example that uses blockchain technology. The digital currency bitcoin itself is highly controversial but the underlying blockchain technology has worked flawlessly and found wide range of applications in both financial and non financial world. The main hypothesis is that the blockchain establishes a system of creating a distributed consensus in the digital online world. This allows participating entities to know for certain that a digital event happened by creating an irrefutable record in a public ledger. It opens the door for developing a democratic open and scalable digital economy from a centralized one. There are tremendous opportunities in this disruptive technology and revolution in this space has just begun. This white paper describes blockchain technology and some compelling specific applications in both financial and non financial sector. We then look at the challenges ahead and business opportunities in this fundamental technology that is all set to revolutionize our digital world.

1.2 PROBLEM DEFINITION

Due to misrepresentation, double-dealing, debasement, dishonesty, kid work, and financial rejection—often carried out by persuasive entertainers—the livelihood of smallholder ranchers in the cocoa production network of developing nations is intolerable. Due to the above-mentioned problem, there is need to develop a web app for smallholder ranchers in cocoa production that resolves those problems. The proposed web application must solve all the problems like misrepresentation financial rejection, etc., and provide more security features and accurate search results to the customers and secure payments for users off the web application. The livelihood of smallholder farmers in emerging economies' supply chain is substandard because of fraud, exploitation, corruption, deceit, child labor, and financial exclusion, usually perpetrated by influential actors. The situation creates a social sustainability problem which needs urgent attention.Digital technologies such as sensors, drones, satellites.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

1.Title: Blockchain technologies for sustainability in the agrifoods sector: a literature review of academic research and business perspectives

Year: 2022 , Authors: F. Dal Mas, Massaro, M. Ndou ,V. Raguseo

In the realm of digital transformation, Blockchain technologies are considered prominent for managing agrifood supply chains due to their contribution to traceability, safety, quality, transparency, and scalability. Both academia and businesses are paying increased attention to the application of Blockchain technologies in the agrifood sector, also highlighting their potential to address sustainable issues.

2.Title: Technology for Social Good Foundations: A Perspective from the Smallholder Farmer in Sustainable Supply Chains

Year: 2021 , Authors: Matthew Quayson, Chunguang Bai, Joseph Sarkis

The livelihood of smallholder farmers in emerging economies' cocoa supply chain is substandard because of fraud, exploitation, corruption, deceit, child labor, and financial exclusion, usually perpetrated by influential actors. This situation creates a social sustainability problem which needs urgent attention.

3.Title: Digital Inclusion For Resilient Post-COVID-19 Supply Chains: Smallholder Farmer Perspectives

Year: 2021, Author: Matthew Quayson; Chunguang Bai; Vivian Osei

The coronavirus (COVID-19) pandemic has and continues to have far-reaching global economic and environmental implications. In developing economies and regions, the pandemic's disruption of the agriculture commodity supply chain has made it difficult for smallholder farmers to exist. Smallholder farmer's traditional struggles have worsened.

4.Title: Toward agent-based models for pre-harvest food safety
Year: 2020, Authors: P. W. Bergholz, M. Wiedmann.

The safety of fresh produce is a concern because many types of produce are consumed raw, and the incidence of produce-borne diseases appears to be increasing. Good agricultural practices (GAPs) can limit the chances for contamination of produce, but farms are open systems where interactions with surrounding lands can override benefits from GAPs. The extent to which interactions between farms and the surrounding landscape contribute to sporadic disease is not yet clear, but outbreaks have arisen from contamination of produce by wildlife, manure, or irrigation water. We hypothesize that the ability to control transmission of pathogens to produce, before harvest, partially depends on understanding the dynamics of pathogen populations, with some landscapes exhibiting combinations of characteristics that increase pathogen dispersal to croplands. The landscape processes leading to contamination are difficult to predict with current techniques. In this perspectives paper, we advance a foundation for modeling food contamination processes in agricultural landscapes using a combination of remote-sensing technologies and ecological modeling of pathogen dispersal, enabled by geographic information systems. With the development of predictive models, cultivation practices can be adapted to local conditions to reduce food contamination by controlling the connectivity between foodborne pathogen reservoirs and food production blocks on farms.

5.Title: Analysis on Argo-Ecological landscapes pattern in Phaeozem land area in Northeast China

Year : 2020, Authors: Zhang Bai, Zhang Xuelin, Ma Chaoqun.

The spatial structure of landscape is the result of complicated interaction among nature conditions, living organisms and human activities, which changes reflect the interrelationship among them. With remote sensing (RS) images in the same season of different years, we can get the generalized and visualized information of landscape and its changes. With the help of Geographic Information System (GIS) software and RS image processing software, we interpreted the images andquantified the information of the landscape and its changes. By overlaying the land-use maps, we identified the location of changes and conversion between different land-use types. Phaeozem (black soil) is the main type of cultivated soil,distributed in the shape of belt in the mesa and terrace in the east of Songnen Plain in Northeast China, Phaeozem is rich in organic matters and very fertile, and the reclamation rate is high, which makes Songnen Plain be the main commodities food base, and contribute a lot to the social and economical development of Northeast China and even the whole country. The continuous reclamation for the forest land and grassland to the cultivated land made the landscape simplify, and expansion of the cities and transportation makes the space for the agriculture development limited, since "market economy" implemented at the beginning in 20th century, the comparative profit of paddy field and dry land drives the conversion to paddy field.

6. Title: Linking NPZD and food web models of an estuarine lagoon ecosystem

Year: 2020, Authors: Ali Erturk, Arturas Razinkovas, Petras Zemlys.

Among aquatic ecosystems, estuarine lagoons are generally more complex than inland waters because of the combined effect of the land and the sea. These systems are under the influence of land through the rivers and have a restricted and temporally variable water exchange with the seas or the oceans. Estuarine lagoons are generally productive ecosystems which offer habitats for many species. Many coastal and estuarine lagoons are known to be important components of the natural capital providing opportunities for aquaculture. These ecosystems are difficult to analyze. Like most of the transitional waters, their trophic and ecological status cannot be defined easily because of their complexity. Most of them are usually under strong human influence or already modified heavily, which makes this task even more difficult. Nutrient phytoplankton zooplankton detritus (NPZD) models can help scientists to analyze the full picture of an aquatic system, together with physical, chemical and biological processes, to fill in the gaps of data between samplings and to forecast environmental changes and use this output for planning. Traditionally, these models have been developed and used by engineers extensively, usually with the aim of water resources and quality management. However, different needs may arise in ecological studies with different aims such as better understanding how an aquatic ecosystem works or analyze the interactions in an aquatic food web. In this case, different state variables may be needed than the conventional ones, which are used by more general water quality models that are available. In ecological studies, related to coastal lagoon ecosystems organism groups on the higher trophic levels can be of interest. In this study, an NPZD model and a trophic network model that contains organism groups on the higher trophic levels were linked using the ldquo bottom-up control rdquo approach.

CHAPTER-3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

To resolve this issue present an effective TBM calculation to perform coordinating inside a sliding window. We likewise propose a DRL-based optimization to additional improve to additional improve the arrangement quality. Broad exploratory outcomes affirm the viability and proficiency of our proposed calculations.

3.1.1 Drawbacks

- SMA is slower to respond to cost changes.
- Here information's not in figure text then there's no any sort of warning.

3.2 PROPOSED WORK

Cryptographic techniques coordinate hypertext, which integrates applying a structure called a calculation to plain text to transform it into something that will transmit an impression of being babble to any individual who doesn't have the strategy for deciphering it. Then, the provider needs to pay and the subtleties of piece receipt can ready to see by the provider and the director

3.2.1Advantages

- Changes and the information will be switched over completely toencode text by utilizing crptography.
- With the assistance of SQL inquiries can get warning from recipient

3.3 REQUIREMENT ANALYSIS AND SPECIFICATION

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

- Functional requirements
- Non-Functional requirements
- Environment requirements
    A. Hardware requirements
    B. Software requirements

3.3.1 Functional Requirements

The software requirements specification is the first step in the requirements analysis process. It lists requirements of a particular software system.

3.3.2 Non-Functional Requirements

Process of non-functional steps,

- Problem definition
- Preparing data
- Evaluating algorithm
- Improving results
- Prediction the result

3.3.3 Environmental Requirements

1. Software Requirements

- Operating System : Windows / Linux Simulation
- Tool : Eclipse with MySQL

2. Hardware Requirements

- Processor : Dual Core 2 Duos
- Hard disk : minimum 250 GB
- RAM : minimum 2 GB DD

3.4 TECHNOLOGY STACK

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it&#39;s the largest and most active on the planet. With its versatilty, efficiency, and portability, Java has becomeinvaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun’s java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

3.4.1 Object Oriented

To be an Object Oriented language, any language must follow at least the four characteristics.

1.Inheritance: It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a features as needed.

2.Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3.Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

4.Dynamic binding: Sometimes we don&#39;t have the knowledge of objects about their specific types while writing our code. It is the way of providingthe maximum functionality to a program about the specific type at runtime.

3.4.2 Java Server Pages

Java Server Pages or JSP for short is Sun&#39;s solution for developing dynamic websites. JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy. JSP pages are efficient, it loads into the web servers memory on receiving the request very first time and the subsequent calls are served within a very short period of time. In today’s environment most web sites servers dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very cc easy to develop database drivenweb application. Java is known for its characteristic of “write once, run anywhere.” JSP pages areplatform Java Server Pages.Java Server Pages (JSP) technology is the Java platform technology for deliveringdynamic content to web clients in a portable, secure and well-defined way. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms.JSP has been built on top of the Servlet API and utilizes Servlet semantics. JSP has become the preferred request handler and response mechanism. Although JSP technology is going to be a powerful successor to basic Servlets, they have an evolutionary relationship and can be used in a cooperative and complementary manner. Servlets are powerful and sometimes they are a bit cumbersome when it comes to generating complex HTML. Most servlets contain a little code that handles application logic and a lot more code that handles output formatting. This can make it difficult to separate and reuse portions of the code when a different output format is needed. For these reasons, web application developers turn towards JSP as their preferred servlet environment.

3.4.3 Evolution of Web Applications

Over the last few years, web server applications have evolved from static to dynamic applications. This evolution became necessary due to some deficiencies in earlier web site design. For example, to put more of business processes on the web, whether in business-to-consumer (B2C) or business-to-business (B2B) markets, conventional web site design technologies are not enough. The main issues, every developer faces when developing web applications, are:

- **Scalability** - a successful site will have more users and as the number of users is increasing fastly, the web applications have to scale correspondingly.
- **Integration of data and business logic** - the web is just another way to conductbusiness, and so it should be able to use the same middle-tier and data-access code.
- **Manageability** - web sites just keep getting bigger and we need some viable mechanism to manage the ever-increasing content and its interaction with business systems.
- **Personalization** - adding a personal touch to the web page becomes an essential factor to keep our customer coming back again. Knowing their preferences, allowing them to configure the information they view.

their past transactions or frequent search keywords are all important in providing feedback and interaction from what is otherwise a fairly one-sided conversation. Apart from these general needs for a business-oriented web site, the necessity for new technologies to create robust, dynamic and compact server-side web applications has been realized. The main characteristics of today's dynamic web server applications are as follows:

- Serve HTML and XML, and stream data to the web client
- Separate presentation, logic and data
- Interface to databases, other Java applications, CORBA, directory and mailservices
- Make use of application server middleware to provide transactional support.
- Track client sessions.

3.4.4Benefits of JSP

One of the main reasons why the JavaServer Pages technology has evolved into what it is today and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. Another benefit of utilizing JSP is that it allows to more cleanly separate the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a number of exciting benefits, which are chronicled as follows:

- The JSP technology is platform independent, in its dynamic web pages, its web servers, and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web enabled application server. The JSP pages can be accessed from any web server.
- The JSP technology emphasizes the use of reusable components. These

components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA WebLogic and IBM WebSphere.

3.4.5 Servlets

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.

3.4.6 Java Servlets:

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable. Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servets can be handled by separate threads within the same process.

Servlets are also portable and platform independent. A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and he HTTP protocols respectively.

The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

CHAPTER-4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define as a standard way to visualize the way a system has been designed. It is quite similar to blue prints used in other fields of engineering.

4.1.1Use case diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

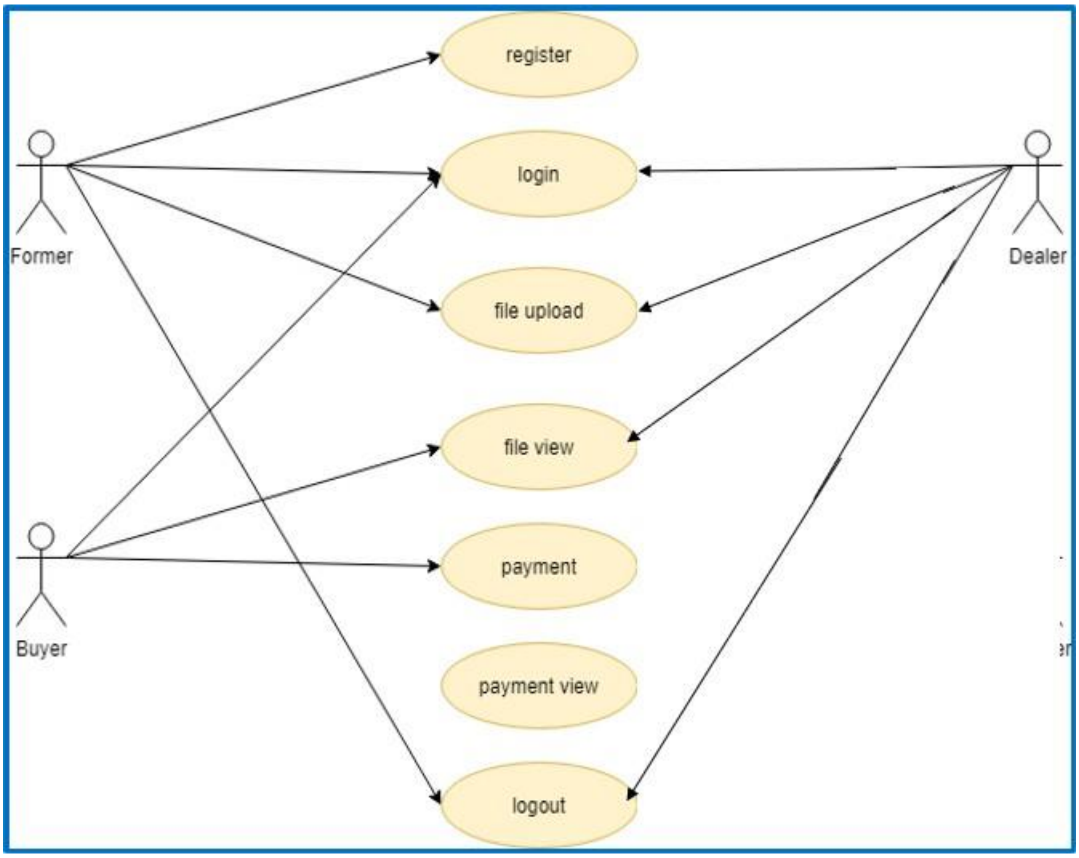


fig4.1 use case diagram

4.1.2 Activity Diagram

A graphical representation of an executed set of procedural system activities and considered as tate chart diagram variation. Activity diagrams describe parallel and conditional activities, usecases and system functions at a detailed level.

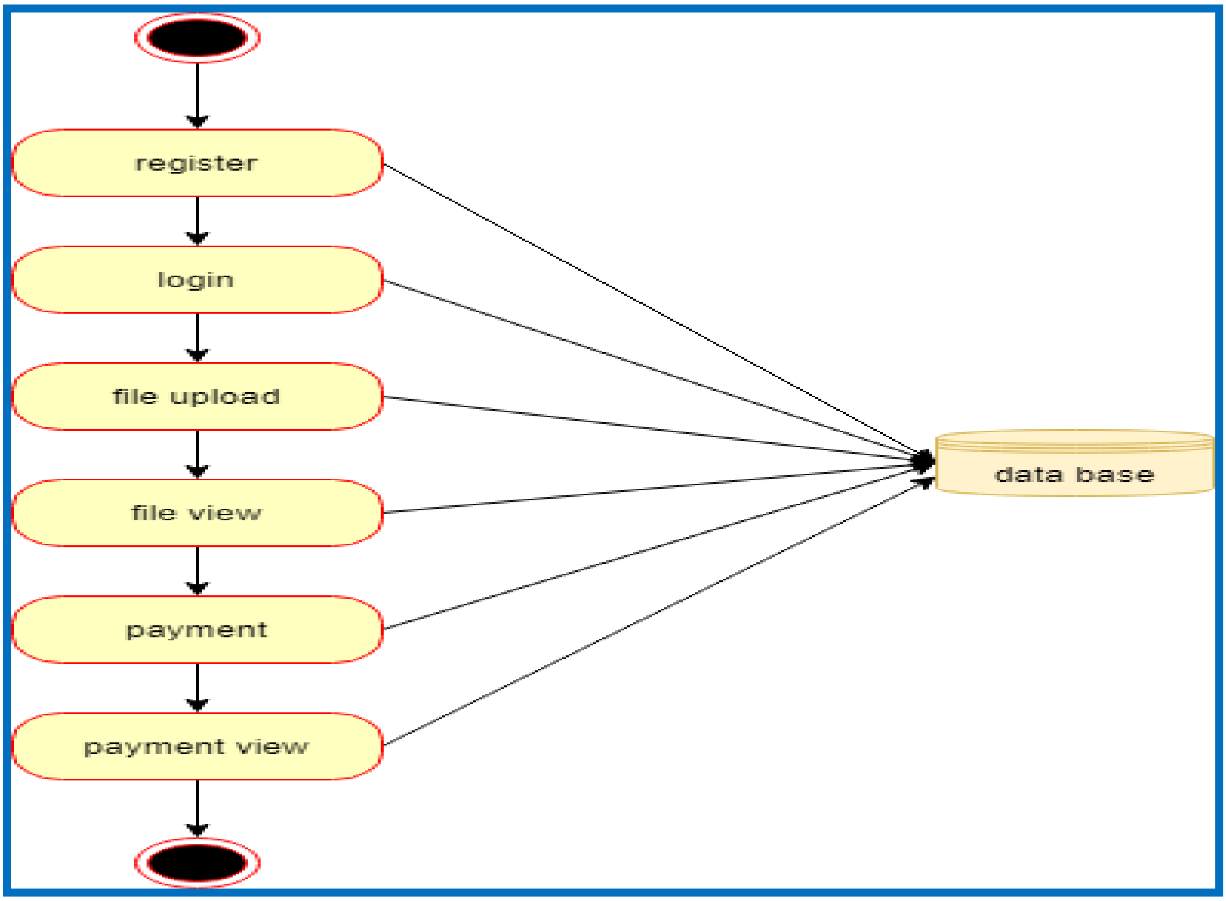


fig 4.1.2activity diagram

4.1.3 Class Diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance. Class diagrams are the most important kind of UML diagram and are vitally important in software development. Class diagrams are the best way to illustratea system's structure in a detailed way, showing its attributes. Attributes describe a value or a range of values that instances of the classifier canhold.

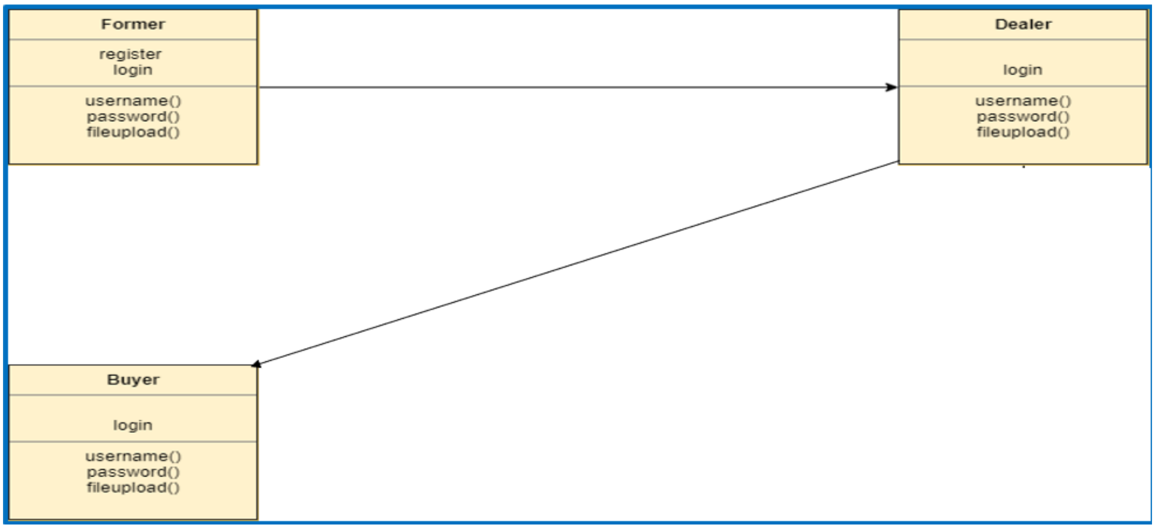


fig 4.1.3 class diagram

4.1.4 Sequence iagram

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. For example, lifelines in a sequence diagram for a banking scenario can represent a customer, bank teller, or bank manager. The communication between the customer, teller, and manager are represented by messages passed between them. The sequence diagram shows the objects and the messages between the objects.

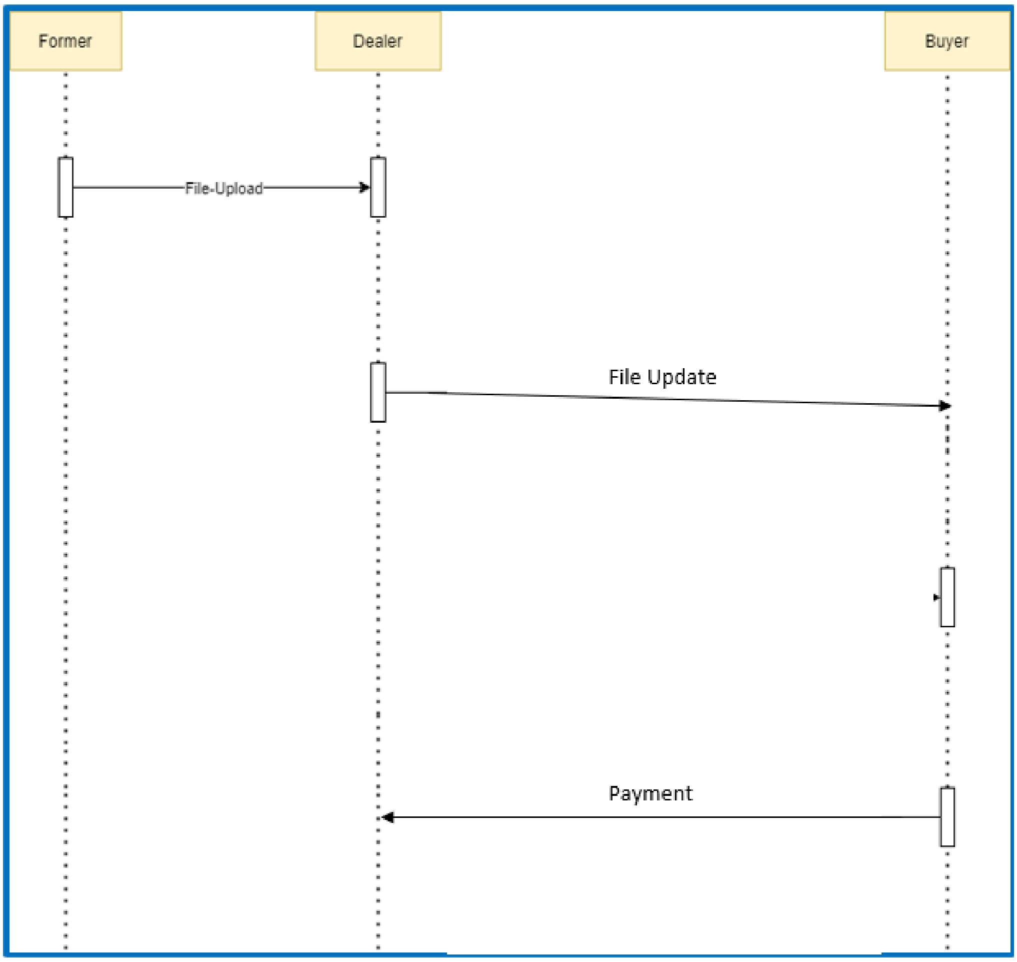


fig 4.1.4 sequence diagram

4.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects .It can be used for the visualization of data processing (structured design).Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. DFD Level 0 is also called a Context diagram. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a one.

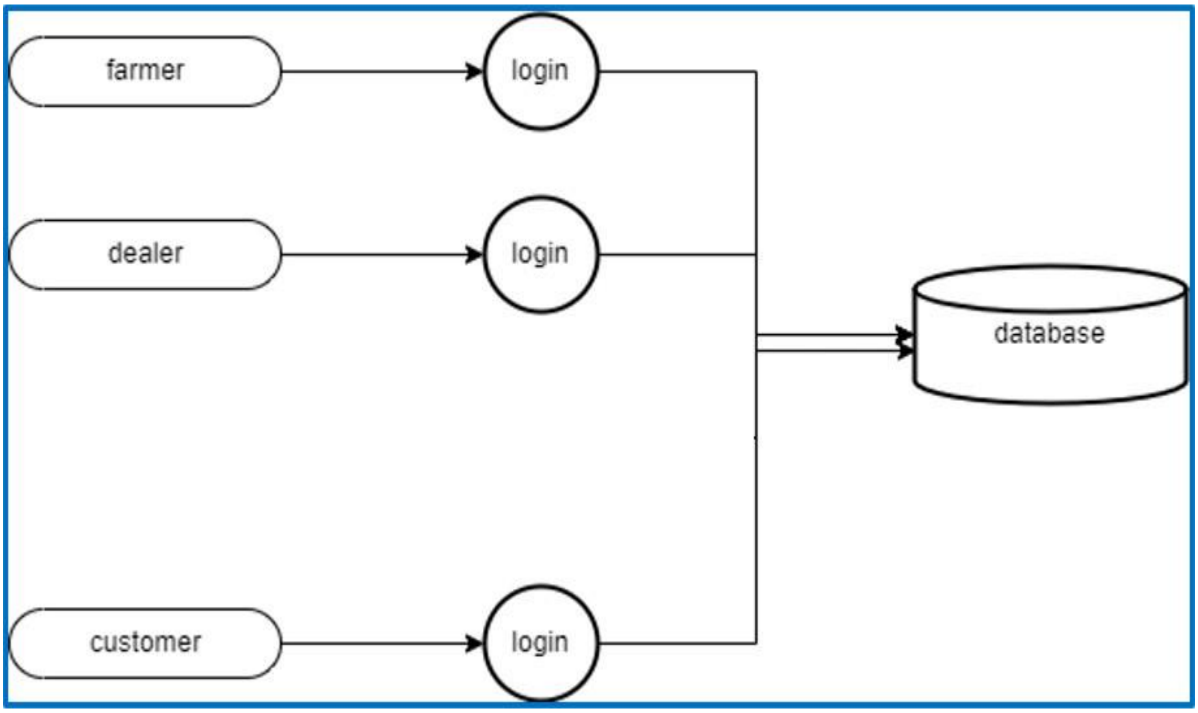


fig 4.2.1 dfd diagram level 0

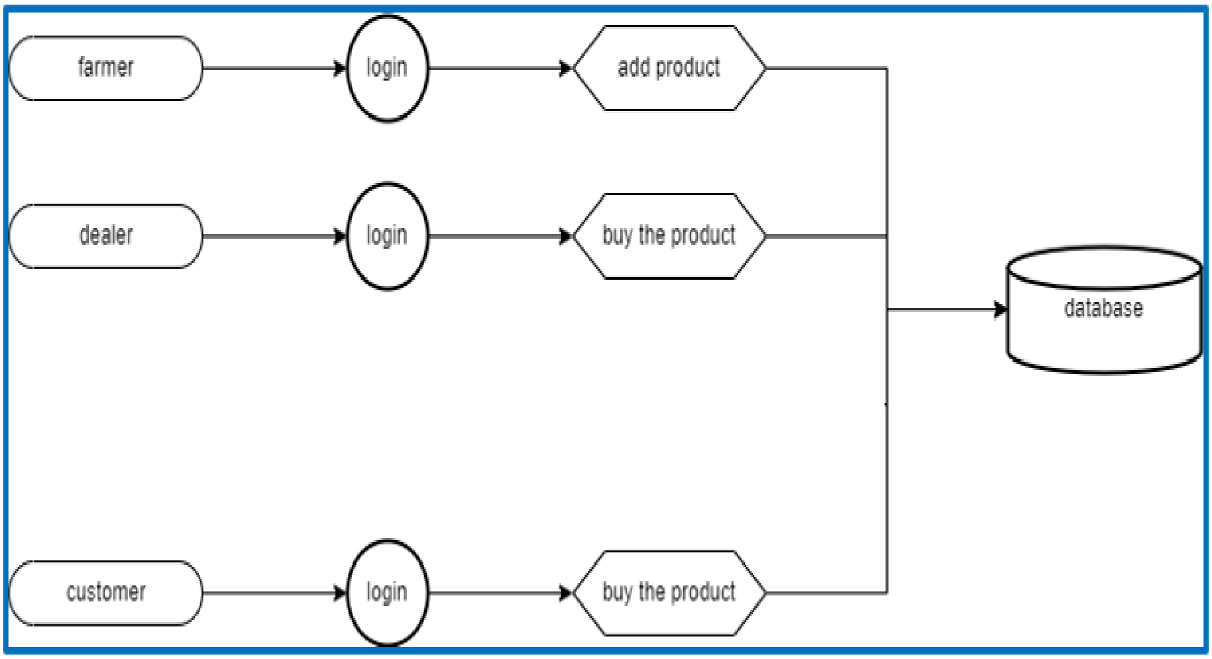


fig 4.2.2 dfd diagram level 1

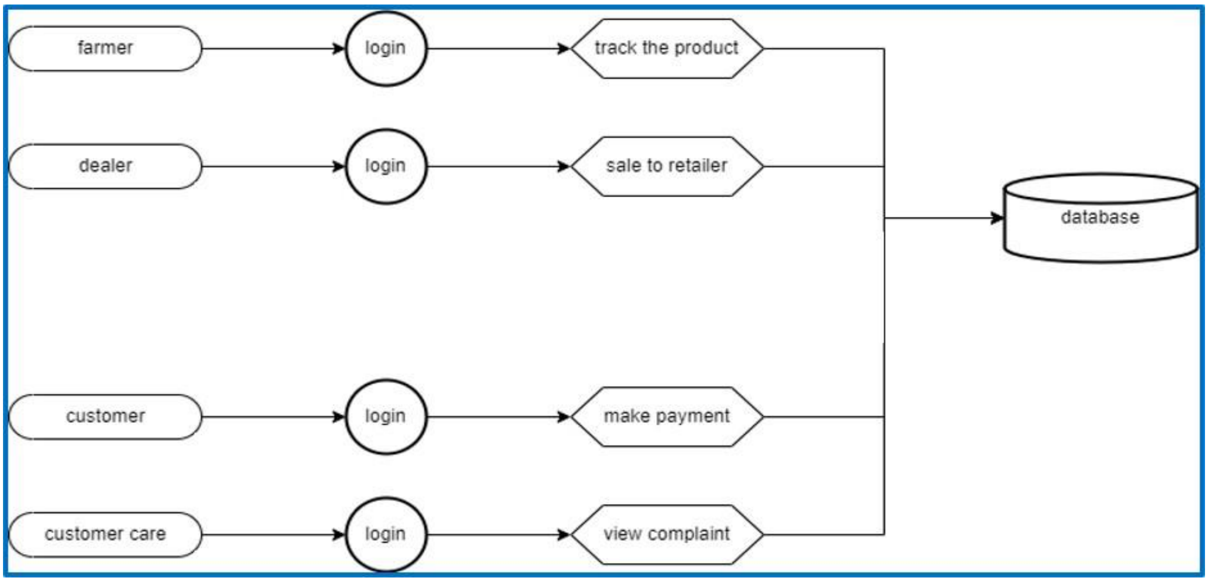


fig 4.2.3 dfd diagram level 2

4.3 ER-DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems. ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It’s also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

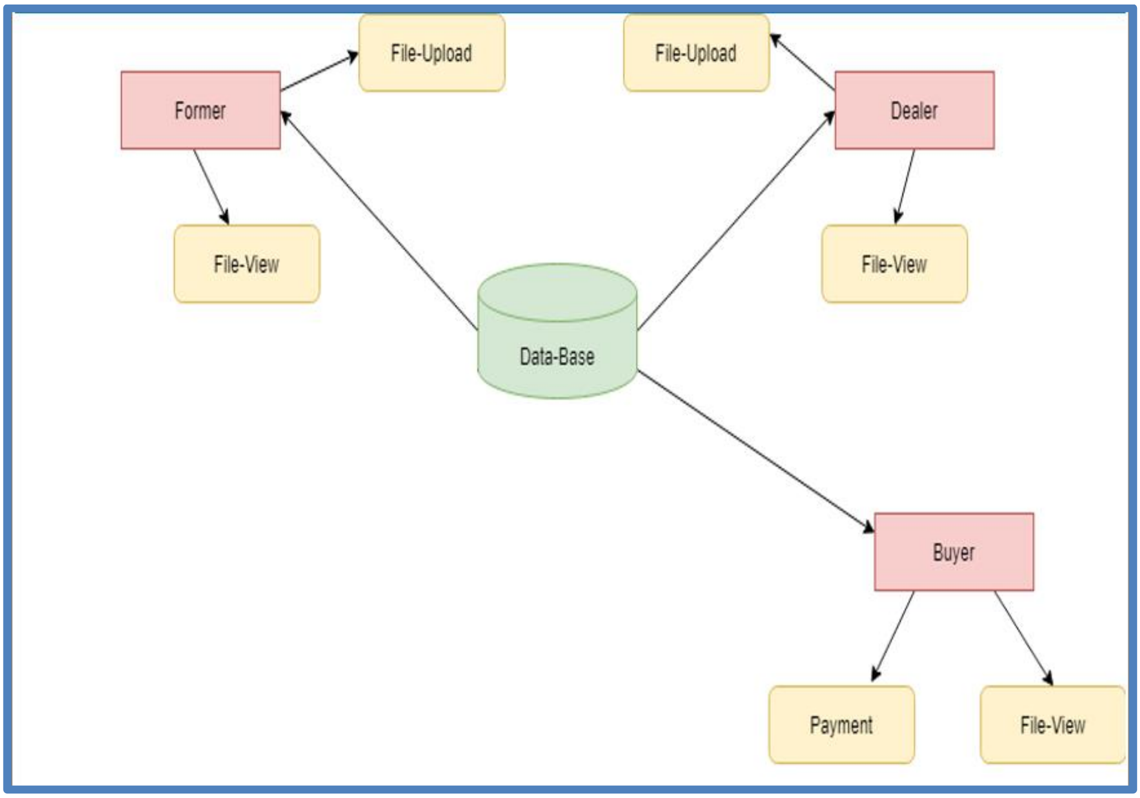


fig 4.3.1 entity-relationship diagram

4.4 DATA-DICTIONARY

A data dictionary provides terminology for all relevant data to be used by the developers in a project. It helps in performing analysis based on the impact of some data on the processing activities. It also helps the developers to determine the definition of different data structures in terms of their basic elements while designing activities. In the case of large systems, data dictionaries may become extremely voluminous and difficult to handle. In such case, CASE (Computer-Aided Software Engineering) tools are used, that capture all the data items appearing in the DFD and automatically generate the data dictionary .Using any convention’s DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

- **External entity**: an outside system that sends or receives data, communicating with the system being diagrammed.
- **Process**: any process that changes the data, producing an output. Data store: files or repositories that hold information for later use, such as a database table or a membership form.
- **Data flow**: the route that data takes between the external entities, processes and data stores

ATTRIBUTE_NAME	**DATA_TYPE**	**DESCRIPTION**
Name	Varchar2	Name of farmer
Email	Varchar2	Email of farmer
Password	Varchar2	Password for login
Crop-type	image	Image for crop

table 4.4.1 farmer new account creation

ATTRIBUTE_NAME	**DATA_TYPE**	**DESCRIPTION**
Name	Varchar2	Name of the buyer
Email	Varchar2	Email of buyer
Password	Varchar2	Password for login

table 4.4.2 customer new account creation

ATTRIBUTE_NAME	**DATA_TYPE**	**DESCRIPTION**
Name	Varchar2	Name of farmer
Email	Varchar2	Email of farmer
Password	Varchar2	Password for login
Crop-type	image	Image for crop

table 4.4.3 dealer new account creation

CHAPTER-5

SYSTEM ARCHITECTURE

5. SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. Three types of system architectures are identified, integrated, distributed and mixed, (partly integrated and partly distributed). It is shown that the type of interfaces defines the type of architecture. Integrated systems have more interfaces, which furthermore are vaguely defined.

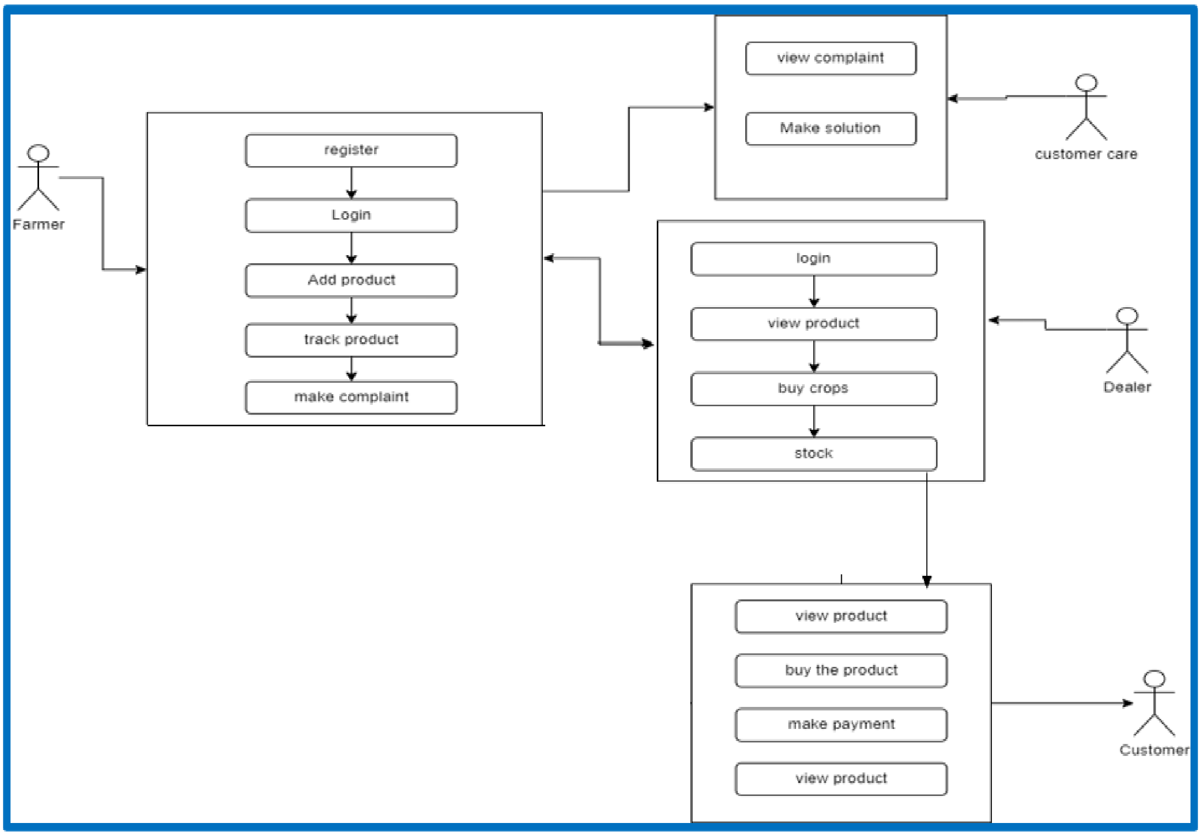


fig 5.1 system architecture

5.2 MODULE DESIGN SPECIFICATION

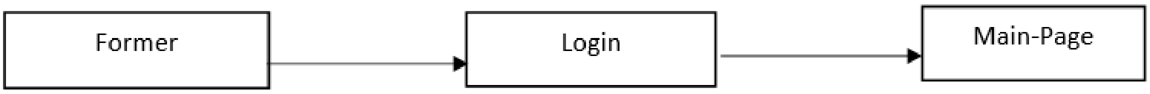
MODULES

- Login
- Farmer file-upload
- Buyer file-view
- Buyer payment
- Farmer paid-view

5.2.1 MODULE DESCRIPTION

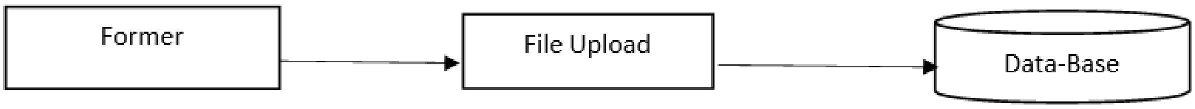
1. LOGIN

This is the first module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.



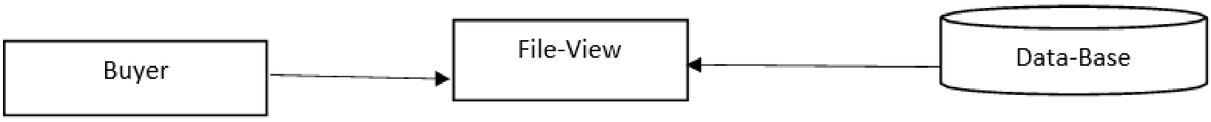
2.FARMER FILE-UPLOAD

In this module is used to help to the farmer to upload the file with the land longitude and the user will update the report along with their opinion and their opinion will be stored in the database.



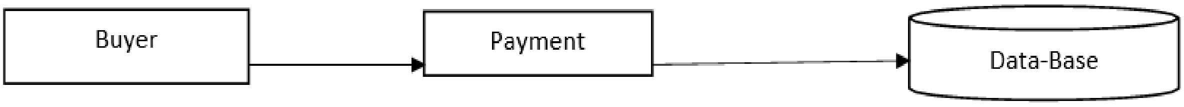
3.BUYER FILE VIEW

In this module the Buyer will also view the data file regarding the crops and analyses the price of the crops and discuss the cost for crops according to the quantity . The Buyer will be responsible for your file stored in database.



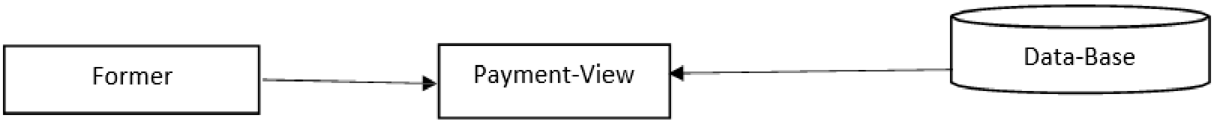
4. BUYER PAYMENT

In this module the Buyer will also view the payment page of the data file and fully analyze the data in category view which helps the Buyer to know whatthey ordered and how much does it costs.



5.FARMER PAYMENT VIEW

In this module the farmer will also view the payment page and analyses the Quantity and amount of the crops sold. The farmer will be responsible for the file to be stored in the database



HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

Basic Construction of an HTML Page

These tags should be placed underneath each other at the top of every HTML page that you create**.**

CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users.CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let's try to break the acronym:

- Cascading: Falling of Styles
- Style: Adding designs/Styling our HTML tags
- Sheets: Writing our style in different documents

CSS Syntax

```
Selector {
          Property 1 : value;
         Property 2 : value;
         Property 3 : value;
           }
```

CHAPTER-6

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

6.1 CODING

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<style>body {
background-image: url("images/reg2.jpg");
background-repeat: no-repeat;
background-size: cover;
}
h2{
color: BLACK;
}
</style>
</head>
<body>
<center>
<br><br><br><br><br>
<h2>Buyer-Register</h2>
<form action="Buyerregservlet" method="post">
```

```
<input type="text" name="username" placeholder="username"
style="width:150px;height:30px;border-radius:8px;text-align:center;"><br><br>
<input type="text" name="phoneno" placeholder="phoneno"
style="width:150px;height:30px;border-radius:8px;text-align:center;"><br><br>
<input type="text" name="email" placeholder="email"
style="width:150px;height:30px;border-radius:8px;text-align:center;"><br><br>
<input type="password" name="password" placeholder="password"
style="width:150px;height:30px;border-radius:8px;text-align:center;"><br><br>
<input type="submit" value="Submit"style="width:100px;height:40px;border-radius:8px;background-color:blue;color:white;">
</form>
</center>
</body></html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<style>body {
background-image: url("images/reg2.jpg");
background-repeat: no-repeat;
background-size: cover;
}
```

```
h2{
color: BLACK;
}
</style>
</head>
<body>
<center>
<br><br><br><br><br>
<h2>Buyer-Register</h2>
<form action="Buyerregservlet" method="post">
<input type="text" name="username" placeholder="username"
style="width:150px;height:30px;border-radius:8px;text-
align:center;"><br><br>
<input type="text" name="phoneno" placeholder="phoneno"

style="width:150px;height:30px;border-radius:8px;text-
align:center;"><br><br>
<input type="text" name="email" placeholder="email"

style="width:150px;height:30px;border-radius:8px;text-
align:center;"><br><br>
<input type="password" name="password" placeholder="password"

style="width:150px;height:30px;border-radius:8px;text-
align:center;"><br><br>
<input type="submit" value="Submit"style="width:100px;height:40px;border-
radius:8px;background-color:blue;color:white;">
</form>
</center>
```

```
</body>
</html>

<%@ page language=“java” contentType=“text/html; charset=ISO-8859-1” pageEncoding=“ISO-8859-1”%>

<%@page import=“dbcon.Dbconn”%>

<%@page import=“java.sql.ResultSet”%>

<%@page import=“java.sql.PreparedStatement” %>

<%@page import=“java.sql.*” %>
<%@page import=“java.util.*” %>
<%@page import=“java.io.FileInputStream” %>

<%@page import=“java.io.IOException” %>

<%@page import=“java.io.PrintWriter” %>

<!DOCTYPE html PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN” “http://www.w3.org/TR/html4/loose.dtd”>
<html>
<head>
<meta http-equiv=“Content-Type” content=“text/html; charset=ISO-8859-1”>
<title>Insert title here</title>
<link rel=“stylesheet” href=“css1/bootstrap.min.css”>
</head>
<%-- <% String cemail=session.getAttribute(“cemail”).toString(); %> --%>
<body>
<center>
<a href=“bankmainpage.jsp”><button type=“button” class=“btn btn-primary”>back</button></a>
```

```
</center>
<div class="container">
<div class="table-responsive-sm">
<table class="table table-striped">
<thead>
<tr>
<th>Charity Name</th><th>Charity mail</th>
<th>People Email</th>
<th>Purpose</th>
<th>Amount</th>
<th>Forword</th>
</tr>
</thead>
<%
Connection con; con=Dbconn.create();
PreparedStatement ps=con.prepareStatement("SELECT * FROM
`scam`.`checkdetails` where status=&#39;Verified&#39; ");ResultSet
rs=ps.executeQuery();
while(rs.next())
{
String pic=rs.getString(10);
%>
<tbody>
<tr>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(8)%></td>
<td><%=rs.getString(13)%></td>
<td><%=rs.getString(6)%></td>
<td><%=rs.getString(3)%></td>
```

```
<%-- <td><p class=“card-text”><img class=“card-img-top”
src=“Local1/<%=pic        %>“   width=“40”  height=“150”        alt=“Card
 imagecap”></p></td>
--%>
<td><a
href=“transferring.jsp?id=<%=rs.getString(1)%>&amp;&amp;chckno=<%=rs.g
etString(2)%>&amp;&amp;
chckword=<%=rs.getString(3)%>&amp;&amp;date=<%=rs.getString(4)%>&a
mp;&amp;charityname=<%=rs.g
etString(5)%>&amp;&amp;desc=<%=rs.getString(6)%>&amp;&amp;amount=
<%=rs.getString(7)%>&amp;&amp;cmail=
<%=rs.getString(8)%>&amp;&amp;city=<%=rs.getString(9)%>&amp;&amp;pic
=<%=rs.getString(10)%>&amp;&amp;e
ncrypt=<%=rs.getString(11)%>&amp;&amp;status=<%=rs.getString(12)%>&a
mp;&amp;pemail=<%=rs.getS
tring(13)%>&amp;&amp;address=<%=rs.getString(14)%>&amp;&amp;bname
=<%=rs.getString(15)%>&amp;&amp;bacc
=<%=rs.getString(16)%>&amp;&amp;bifsc=<%=rs.getString(17)%>“><button
class=“btn btn-
primary”>Transfer</button></a>
<th>Charity mail</th>
<th>People Email</th>
```

```
<th>Purpose</th>

<th>Amount</th>

<th>Forword</th>

</tr>

</thead>

<%Connection con; con=Dbconn.create();
PreparedStatement ps=con.prepareStatement("SELECT * FROM
`scam`.`checkdetails` where status=&#39;Verified&#39; ");ResultSet
rs=ps.executeQuery();
while(rs.next()){
String pic=rs.getString(10);
%>
<tbody>
<tr>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(8)%></td></tr>
</tbody>
<%} %>
</table>
</div>
</div>
</body>
</html>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
```

```
<%@page import="dbcon.Dbconn"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.PreparedStatement" %>

<%@page import="java.sql.*" %>

<%@page import="java.util.*" %>

<%@page import="java.io.FileInputStream" %>

<%@page import="java.io.IOException" %>

<%@page import="java.io.PrintWriter" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" href="css1/bootstrap.min.css"></head>
<%-- <% String cemail=session.getAttribute("cemail").toString(); %> --%>
<style>body{
background-color:#8BA6E9;
}
table{
background-color:#D7B7BC;
}
</style>
<body><br><br><br>
```

```
<center>
<button type=“button” class=“btn btn-primary”
onclick=“history.back()”>back</button>
</center>
<br>
<div class=“container”>
<div class=“table-responsive-sm”>
<table class=“table table-success table-striped”>
<thead class=“table-dark”>
<tr>
<th>Meassage</th>
<th>Charity mail</th>
<th>People Email</th>
<th>Charity Name</th>
<th>Deactivate Account</th>
</tr>
</thead>
<%Connection con; con=Dbconn.create();
PreparedStatement ps=con.prepareStatement(“SELECT * FROM
`scam`.`fakecheck` where status=&#39;complaints&#39; “);ResultSet
rs=ps.executeQuery();
while(rs.next()){
%>
<tbody>
<tr>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(7)%></td>
<td><%=rs.getString(11)%></td>
<td><%=rs.getString(4)%></td>
<td><a href=“Complaintcharlist.jsp?id=<%=rs.getString(7)%>“><button
class=“btn btn-primary”>Charity List</button></a>
</tr>
</tbody>
<%} %>
</table>
</div>
</div>
</body>
</html>
<%@ page language=“java” contentType=“text/html; charset=ISO-8859-1”
pageEncoding=“ISO-8859-1”%>
<%@ page import=“java.util.*”%>
<%@page import=“dbcon.Dbconn”%>
<%@page import=“java.sql.ResultSet”%>
<%@page import=“java.sql.PreparedStatement” %>
<%@page import=“java.sql.*” %>
<%@page import=“java.util.*” %>
<%@page import=“java.io.FileInputStream” %>
<%@page import=“java.io.IOException” %>
<%@page import=“java.io.PrintWriter” %>
<head>
<script type=“text/javascript”
src=“https://github.com/rubyeffect/easy_fill/tree/master/lib/easy_fill-
0.0.1.min.js”></script>
<title>Money Transfer</title>
<link rel=“stylesheet” type=“text/css” href=“appointment.css” />
```

```
<link rel="stylesheet" href="css1/bootstrap.min.css">

<link rel="stylesheet" href="css/font-awesome.min1.css">

</head>

<% String id=request.getParameter("id");

String checkno=request.getParameter("chckno"); String
checkwrd=request.getParameter("chckword");String
date=request.getParameter("date");
String charityname=request.getParameter("charityname");
String desc=request.getParameter("desc");
String amount=request.getParameter("amount");
String cmail=request.getParameter("cmail");
String city=request.getParameter("city");
String pic=request.getParameter("pic");
String encrypt=request.getParameter("encrypt");String
status=request.getParameter("status"); String
pemail=request.getParameter("pemail"); String
address=request.getParameter("address");
String bname=request.getParameter("bname");
String bacc=request.getParameter("bacc");
String bifsc=request.getParameter("bifsc");
%>
<Style>body{
background-image:url("image/r11.jpg");background-size:cover;
}
</Style>
<body>
```

```
<% Connection con;
String toacc="";
String tobank="";
String toifsc="";
con=Dbconn.create();
PreparedStatement ps=con.prepareStatement("SELECT * FROM
`scam`.`checkdetails` where checkno=&#39;"+checkno+"&#39; "); ResultSet
rs=ps.executeQuery();
while(rs.next())
{
toacc=rs.getString(16);
tobank=rs.getString(15);
toifsc=rs.getString(17);
}%>
<div class="container">
<h3 style="font-weight: bold; text-align: center;margin-top: 10px">Money
Details</h3><br>
<form method="post" action="moneysending.jsp" >
<div class="row">
<div class="card col-sm-5" style="padding-bottom:10px;background- color:
#333;color:#fff;padding:10px;margin-left:30%;width: 38em;margin-bottom:
25px">
<div class="form-group">
<label for="Pemail" style="color:#fff;"><i class="fa fa-inr"></i>Money
From.</label>
<input type="text" class="form-control" id="from"
value="<%=pemail%>"placeholder="Enter price for year" name="from"
required>
</div>
```

```
<div class=“form-group”>
<label for=“Cemail” style=“color:#fff;”><i class=“fa fa- inr”></i>Money
To.</label>
<input type=“text” class=“form-control” id=“to”
value=“<%=cmail%>“placeholder=“Enter price for year” name=“to” required>
</div>
<div class=“form-group”>
<label for=“Cemail” style=“color:#fff;”><i class=“fa fa-inr”></i>To
Account</label>
<input type=“text” class=“form-control” id=“to”
value=“<%=toacc%>“placeholder=“Enter price for year” name=“to” required>
</div>
<div class=“form-group”>
<label for=“Cemail” style=“color:#fff;”><i class=“fa fa-inr”></i>To
Bank</label>
<input type=“text” class=“form-control” id=“to”
value=“<%=tobank%>“placeholder=“Enter price for year” name=“to” required>
</div>
<div class=“form-group”>
<label for=“Cemail” style=“color:#fff;”><i class=“fa fa-inr”></i>To IFSC
</label>
<input type=“text” class=“form-control” id=“to”
value=“<%=toifsc%>“placeholder=“Enter price for year” name=“to” required>
</div>
<div class=“form-group”>
<label for=“Check No” style=“color:#fff;”><i class=“fa fa-inr”></i>Check
No.</label>
<input type=“text” class=“form-control” id=“phone_no”
value=“<%=checkno%>“placeholder=““ name=“checkno” required
```

```
readonly></div>
<div class="form-group">
<label for="Amount" style="color:#fff;">Amount In Word</label>
<input type="text" class="form-control" id="address"
value="<%=checkwrd%>"placeholder="" name="aword" required>
</div>
<div class="form-group">
<label for="date" style="color:#fff;">Date </label>
<input type="text" class="form-control" id="password1" value="<%=date%>"
placeholder="" name="date" required>
</div>
<div class="form-group">
<label for="Amount" style="color:#fff;">Amount</label>
<input type="text" class="form-control" id="phone_no"
value="<%=amount%>" placeholder="" name="amount" required>
</div>
<br>
<center><button class="btn btn-primary btn-block"
style="width:50%;"onclick="return Validate()";>Submit</button></center><br>
</div>
</div>
<hr></form>
</div>
</body>
</html>
```

CHAPTER-7
SYSTEM TESTING

7.SYSTEM TESTING

7.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the softwaredevelopment life cycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.Depending on the organization's expectations for software development, unit testing might include static code analysis, data-flow analysis, metrics analysis, peer code reviews, code coverage analysis and other software testing practices.

7.1.1 White Box Testing

White Box testing is a test case design method that uses the control structure of the procedural design to the drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) verifies the internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system (the source code), as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g., in-circuit testing (ICT).Decision coverage, which reports on whether both the True and the False branchof a given test has been executed100% statement coverage ensures that all code paths or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly. Pseudo-tested functions and methods are those that are covered but not specified (it is possible to remove theirbody without breaking any test case).

7.1.2 Black Box Testing

Black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it.Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional. Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations. One advantage of the black box technique is that no programming knowledge is required.

- Black box testing is done to find incorrect or missing function
- Interface error
- Performance error
- Error in external database access

7.2 INTEGRATION TESTING

Integration testing is a systematic technique for construction of the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A smallset of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach arethat, interface dispenses can be easily found and corrected.The major error that was faced during the project is linking error.

7.3 TEST CASES AND REPORTS

SI NO	ACTIONS TO BE PERFORMED	EXPECTED RESULTS	ACTUAL RESULTS	RESULTS
1.	Selecting "FARMER" button	Open Farmer Login	As expected	pass
2.	Selecting "DEALER" button	Open Dealer Login	As expected	Pass
3.	Selecting "CUSTOMER" button	Open Customer Login	As expected	Pass
4.	Selecting "CUSTOMER CARE" button	Open Customer Care Login	As expected	Pass
5.	Selecting "View" button	Open Product View Page	As expected	Pass
6.	Selecting "PRODUCT UPLOAD" button	Open Product Upload Form	As expected	Pass
7.	Selecting "SHOP" button	Open Shop Page	As expected	Pass
8.	Selecting "PRODUCT VIEW" button	Open Upload Product Page	As expected	Pass
9.	Selecting "RETAILER REQUEST VIEW" button	Open Retailer Request Page	As expected	Pass

CHAPTER-8
CONCLUSION

8. CONCLUSION

8. 1 CONCLUSION

In this project, we have proposed for real-time provisioning in cloud-based industrial applications. It aimed at getting an optimized execution cost dynamically with the consideration of user-defined security configuration constraints. In this process picker hub receive the client parcel and send delivery address everything is monitoring in and finally branch manager send the daily picking parcel and delivery in branch hub every details to send the head office. The branch manager report will send secure the head office. Report will be encrypted first using AES algorithm.

8.2 FUTURE ENHANCEMENT

- Further improvement on the network's accuracy and generalization can be achieved through the following practices. The first one is to implement a real- world database system.

- Using batch optimization is more suitable for larger datasets. Another technique is to improve the efficiency of protocols, in terms of number of messagesexchanged and in terms of their sizes , as well.

- Finally, using two or more algorithm and large database , the website can be usedmore efficiently . Thus, they should be addressed in future research on this topic.

APPENDICES

OUTPUT SCREENSHOTS

This is first page of the web application.

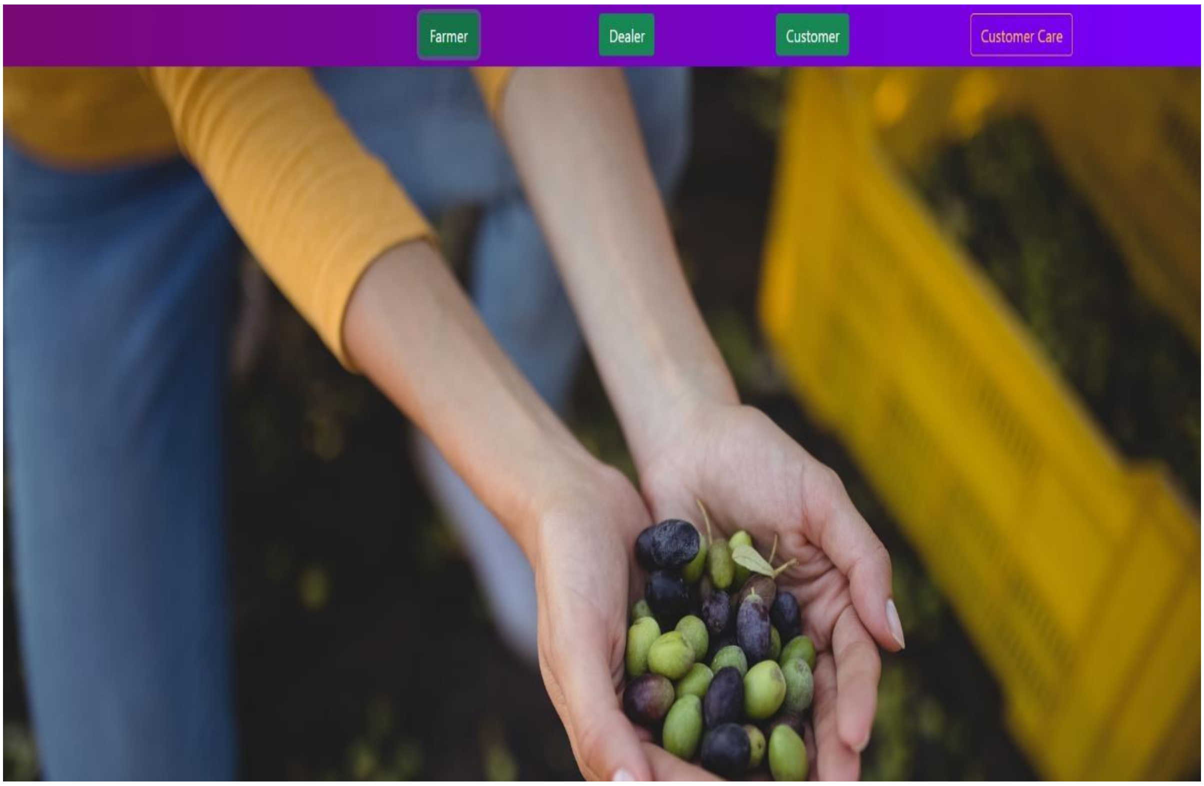


fig a.1 index page

After clicking the Farmer button.

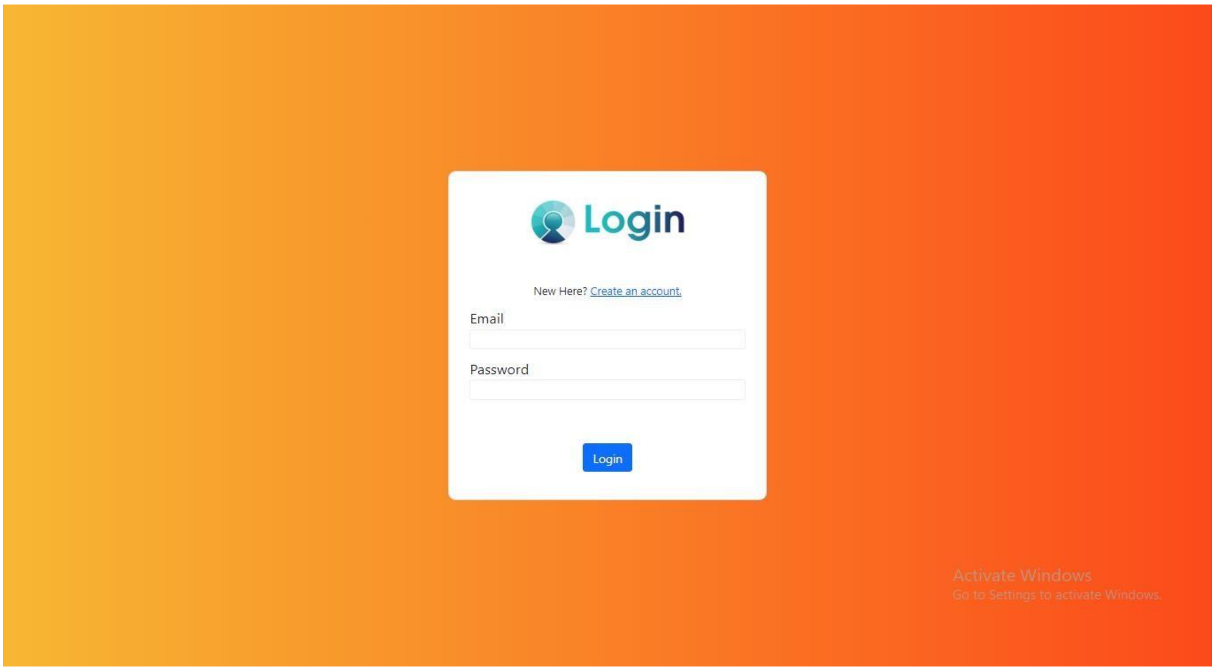
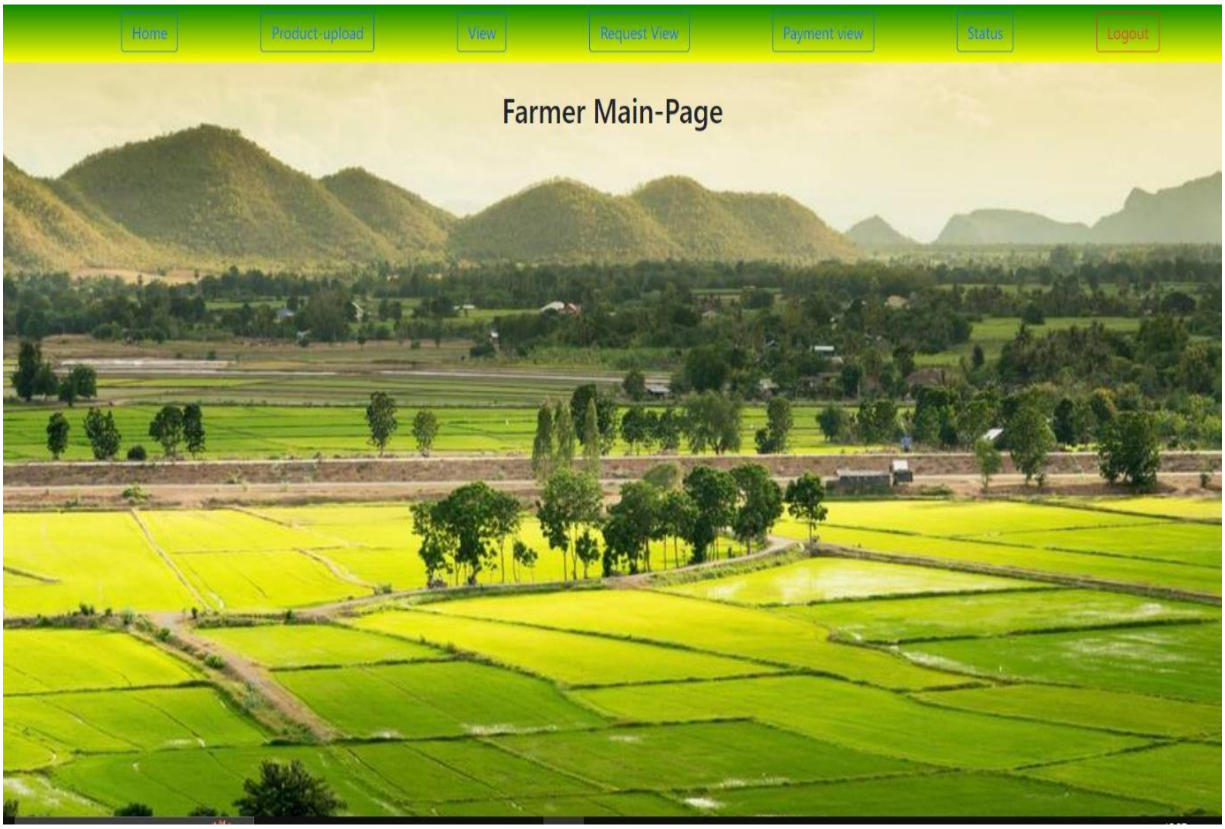

Login
New Here? Create an account.
Email
Password
Login
Activate Windows
Go to Settings to activate Windows.

fig a.2 login page

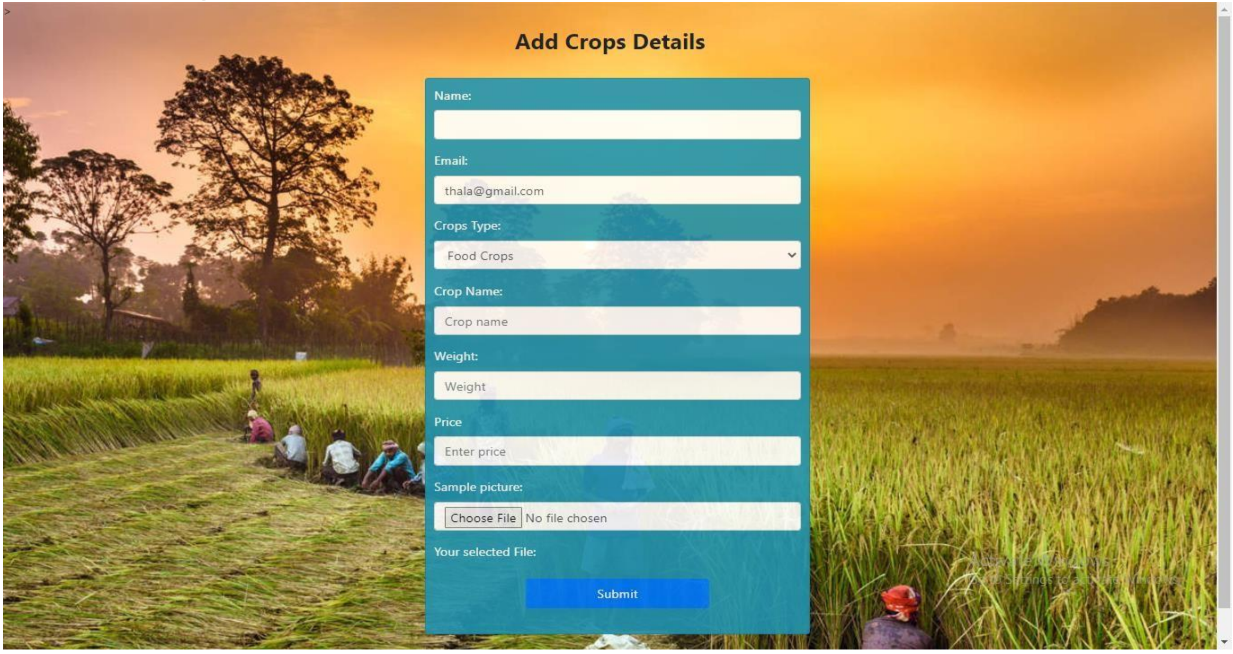
After the successful login of the farmer.



Farmer Main-Page

fig a.3 farmer’s main page

After clicking the product upload button.



Add Crops Details

Name:

Email:
thala@gmail.com

Crops Type:
Food Crops

Crop Name:
Crop name

Weight:
Weight

Price
Enter price

Sample picture:
Choose File No file chosen

Your selected File:

fig a.4 add crops page

After the successful login of the Dealer.

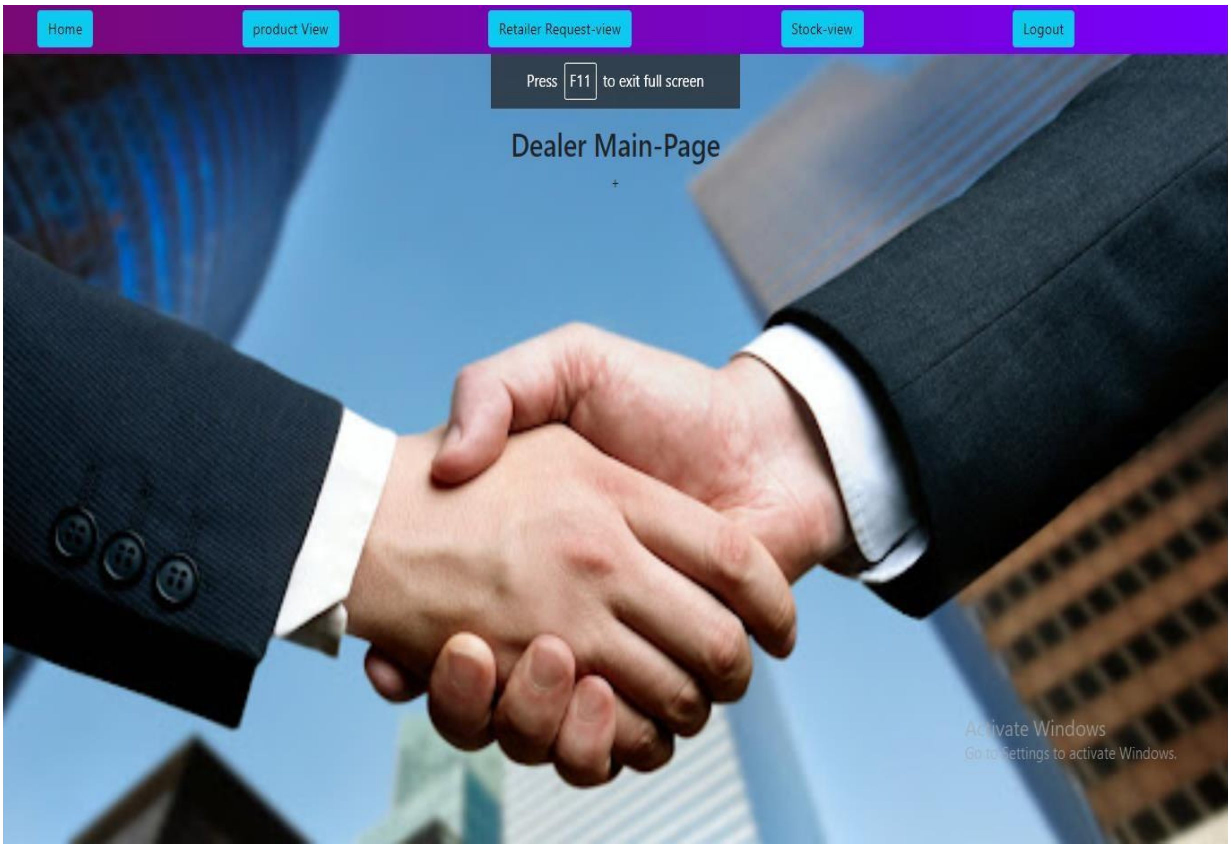


fig a.5 dealer main page

After clicking the Create new account for farmer button.

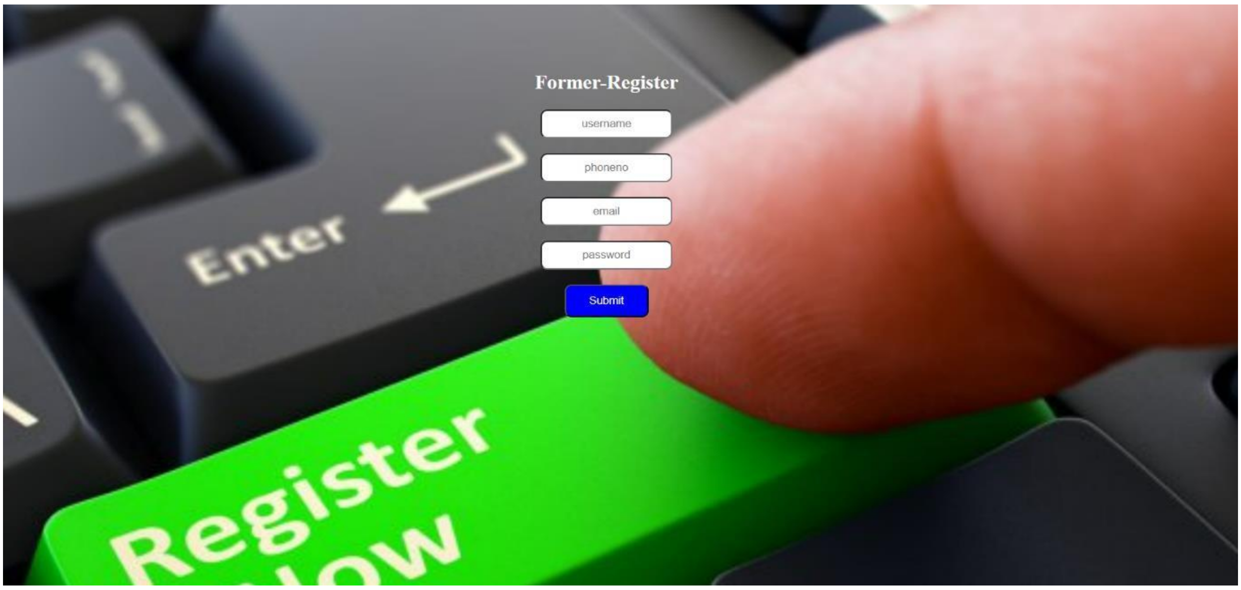


fig a.6 farmer's registration page

REFERENCES

REFERENCES

[1] M. Janssen, V. Weerakkody, E. Ismagilova, U. Sivarajah and Z. Irani, "A framework for analyzing blockchain technology adoption: Integrating institutional market and technical factors", Int. J. Inf. Manage., vol. 50, pp. 302-309, 2020.

[2] A. Kamilaris, A. Fonts and F. X. Prenafeta-Boldύ, "The rise of blockchain technology in agriculture and food supply chains", Trends Food Sci. Technol., vol. 91, pp. 640-652, 2019.

[3] B. Amuzu-sefordzi, K. Martinus, P. Tschakert, and R. Wills, "Disruptive innovations and decentralized renewable energy systems in Africa: A socio-technical review," Energy Res. Social Sci., vol. 46, pp. 140–154, 2018, doi: 10.1016/j.erss.2018.06.014.

[4] C. Bai and J. Sarkis, "Improving green flexibility through advanced manufacturing technology investment: Modeling the decision process," Int. J. Prod. Econ., vol. 188, pp. 86–104, 2017, doi: 10.1016/j.ijpe.2017.03.013.

[5] C. Bai and J. Sarkis, "A supply chain transparency and sustainability technology appraisal model for blockchain technology," Int. J. Prod. Res., vol. 58, no. 7, pp. 2142–2162, 2020, doi: 10.1080/00207543.2019.1708989.

[6] C.Bai, B. Shi, F. Liu, and J. Sarkis, "Banking creditworthiness: Evaluating the complex relationships," Omega, vol. 83, pp. 26–38, 2019.

[7] S. Baurzhan and G. P. Jenkins, "Off-grid solar PV: Is it an affordable or appropriate solution for rural electrification in Sub-Saharan African countries?" Renewable Sustain.EnergyRev., vol. 60, pp. 1405–1418, 2016, doi: 10.1016/j.rser.2016.03.016.

[8] R. Birner and D. Resnick, “The political economy of policies for smallholder agriculture,”World Develop., vol. 38, no. 10, pp. 1442–1452, 2010, doi: 10.1016/j.worlddev.2010.06.001.

[9] C. Andrew, “Modern slavery as a management practice: Exploring theconditions and capabilities for human exploitation,” Acad. Manage. Rev., vol. 38 , no. 1, pp. 45–69, 2013, doi: 10.2307/23416302.

[10] R. Glavee-Geo, U. Burki, and A. Buvik, “Building trustworthy relationships with smallholder (small-scale) agro-commodity suppliers: Insights from the Ghana cocoa industry,” J. Macromarketing, vol. 40, no. 1, pp. 110–127, 2020, doi:10.1177/0276146719900370.

[11] J. H. Grimm, J. S. Hofstetter, and J. Sarkis, “Exploring sub-suppliers' compliance with corporate sustainability standards,” J. Cleaner Prod., vol. 112, pp. 1971–1984, 2016, doi: 10.1016/j.jclepro.2014.11.036.

[12] [Online]. Available: https://smartvillage.ieee.org/our-technology/

[13] M. Janssen, V. Weerakkody, E. Ismagilova, U. Sivarajah, and Z. Irani, “A framework for analyzing blockchain technology adoption: Integrating institutional, market and technical factors,” Int. J. Inf. Manage., vol. 50, pp. 302–309, 2020, doi:10.1016/j.ijinfomgt.2019.08.012.

[14] A. Kamilaris,A. Fonts, and F.X. Prenafeta-Bold´υ, “The rise of blockchain technology in agriculture and food supply chains,” Trends Food Sci. Technol., vol. 91, pp. 640–652, 2019, doi: 10.1016/j.tifs.2019.07.034.

[15] P. Kittipanya-ngam and K. H. Tan, “A framework for food supply chain digitalization: Lessons from Thailand,” Prod. Planning Control, vol. 31, no. 2/3, pp. 158–172, 2020, doi: 10.1080/09537287.2019.1631462.