Green Tech – Agro

Infanta Aalvina J, Kowsika M, Vinodini S

Project Guide : Dr S Prakash

Co-Ordinator: Mrs N G Dharaniya

Department Of Information Technology

Bachelor of Technology

Sri Shakthi Institute of Engineering and Technology (Autonomous), Coimbatore 641 062

ABSTRACT

Green Tech Agriculture revolutionizes traditional farming practices by integrating technology and sustainable methods to benefit farmers and consumers alike. This project focuses on enabling farmers to directly sell their produce to consumers, eliminating intermediaries and extra charges. Leveraging market analysis and consumer demand insights, the project prioritizes cultivating products that are in high demand, thus ensuring profitability for farmers while meeting consumer needs effectively.

Through the implementation of innovative technologies such as online platforms, mobile applications, and data analytics, farmers gain direct access to consumers, bypassing costly middlemen. This direct-selling approach not only increases farmers' profit margins but also fosters transparency and trust in the agricultural supply chain.

Overall, Green Tech Agriculture empowers farmers by providing them with the tools, knowledge, and market insights necessary to thrive in a competitive environment. By eliminating extra charges and focusing on products in demand, this project facilitates a more equitable and sustainable agricultural ecosystem, benefiting both farmers and consumers alike.

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Green Tech-Agro is a web application developed to help the farmers to sell their products to different cities through online.

This Web application is full based on easy way of Agricultural commodities Sales and Purchase and this application contains three modules like Farmer Registration and Login, Common User Registration and Login for all the end user, so the role of the farmer in this application will be that he wants to register all his details including his crop cultivations and then he can directly post his commodities that includes rate, offer etc... Like that each farmer can post their commodities. Common user can purchase the commodities based on the farmer posts so it makes direct selling of product from Agricultural Land itself. So, the Main dashboard contains relevant posts as well as purchase button. By clicking the purchase button Message Order will be sent to the particular farmer automatically. Green Tech-Agro is a project developed to build a website which will help farmers from to sell their products to different cities through online.

- Green Tech-Agro application gives an idea to the farmers how to use Green Tech-Agro to sell their products.
- Farmers will get all the new ideas to improve their productivity and they can buy and sell their products online.
- If the farmers have knowledge of computer then they can directly register in the site and sell their product otherwise they can contact company's computer professional who will schedule classes to teach them basics of computers and internet.
- They can know how they can open this site and register with it and sell their products online etc.
- Green Tech-Agro is a project developed to build a website which will help farmers to sell their products to different cities through online.
- Farmers can use this facility and can learn how it is possible and how they can use Green Tech-Agro to sell their products.
- The green technology policy to provide direction and motivation to continuously enjoy good quality and a healthy environment should be based on four pillars:
- Energy: Seek to attain energy independence and promote efficient utilization.
- Environment: Conserve and minimize the impact on the environment.
- Economy: Enhance the national economic development through the use of technology.
- Social: Improve the quality of life for all

1.2 MODULE DESCRIPTION

The module is to ensure the objectives those are :

- Introduce the concept of green technology, emphasising its purpose and increasing importance in today's global environment.
- Describe sustainable energy generating methods, as well as potential solutions, such as electronic gadgets, for monitoring, modelling, and conserving the natural environment and resources, as well as mitigating the negative effects of human participation.

1) System Independence: This refers to a technical device's capacity to do the needed task on its own. To determine if the technology will demand more capital or labour, the system independence of the technology will be examined.

2) Modernity Image: People should see themselves as contemporary by embracing technology. The message is that individuals are realizing that using a technical item may improve one's social standing while also meeting a basic human need. The image of modernity necessitates that those who embrace it either improve or maintain their social rank.

3) Individual Technology vs Collective Technology: It is the criteria to look at the societal/cultural norms under which the technology functions. To put it another way, it is the meticulous evaluation of technology that is based on a group approach and becomes increasingly system reliant. More systems-independent technology will be required in a society orientated around the individual or single-family unit.

1.3 SYSTEM SPEICFICATION

1.3.1 HARDWARE SPECIFICATION

Processor : INTEL(R)2.10GHz
Installed memory (RAM) : 4 GB
Hard Disk : 160 GB
Operating System : Windows (11)

1.3.2 SOFTWARE SPECIFICATION Front End : XML

Back End : JAVA
Tool : Android studio
Database : SQ Lite

SOFTWARE FEATURES

About JAVA

The purpose of Java in Android application development lies in its ability to provide a robust, versatile, and platform-independent programming language that facilitates the creation of complex mobile applications.

Two key benefits of using Java in Android application development:

1. Platform Independence: Java is known for its "write once, run anywhere" mantra. This means that code written in Java can be executed on any platform that supports Java Virtual Machine (JVM). In the case of Android, Java code is compiled into bytecode, which is then executed on the Android Runtime (ART).
2. This platform independence allows developers to write code once and deploy it across multiple devices running on the Android operating system, simplifying the development process and reaching a broader audience.

2. Rich Ecosystem and Libraries: Java has a vast ecosystem of libraries, frameworks, and tools that streamline Android app development.

1. The Android SDK itself is built on Java, providing developers with APIs for building user interfaces, accessing device hardware features, managing data, and much more.
2. Additionally, Java's extensive collection of third-party libraries, such as Retrofit for networking, Gson for JSON parsing, and Room for local database storage, enables developers to accelerate development and integrate advanced functionality into their applications.

3. Java serves as the foundation for Android app development, offering platform independence, a rich ecosystem of libraries, and a familiar programming language for developers, making it a preferred choice for building robust and feature-rich Android applications.

Why JAVA…

JAVA is consistently rated as one of the world's most popular programming languages. In this section, we will discuss **what makes Java so popular** and what are the major industries that uses Java programming language. The following features make the Java programming language special and popular.

We can see the description of the features in detail. The information helps to get knowledge about the Java programming language.

- Simple to use
- Built-in Security
- Open Source
- Robust API
- Strong community
- Excellent documentation
- Powerful set of Programming Tools
- Versatility

Easy to use

Java's syntax is similar to that of English. The ideal language for beginning Java students who can master the language in two stages.

It starts with Core Java and then moves on to Advanced Java. As a result, Java has a simple learning curve.If we learn Java, then other programming languages are easy to understand. If we already know C and C++, we will be able to learn Java in no time.

Built-in Security

Java programming language uses for creating software applications, Web applications, Android applications, and software tools. The software tools such as Eclipse, NetBeans IDE, IntelliJ IDEA, etc.

Java uses cases have now full-fledged to include the Internet of Things (IoT), Data Science, and Machine Learning applications. It is the reason developers and programmers prefer Java programming language to build applications. It includes high-level concurrency tools and packages also take care of security. The Java classes and related tools create highly scalable solutions for the enterprise

Open Source

To begin, Java is a popular programming language that provides a wide range of app creation capabilities. It is open-source, which implies that it is free to use. Any developer with a working knowledge of Java can use this free platform to create apps.

The Java Development Kit is likewise free to use for all levels of users, from beginners to experts. We utilize two JDK versions: Oracle JDK and the most recent OpenJDK.

Collection of Rich API

Although Java has 52 keywords, its Application Programming Interface (API) is both vast and rich.

The Java API has methods to accomplish all types of tasks, such as networking, connecting to databases, processing XML, handling input-output, etc.

The open-source libraries such as Apache POI, Google Guava, Apache Commons, Apache Xerxes, OpenCV, Gson, can be easily integrated with Java.

Strong Community Support

The main reason for Java's popularity is its active and supportive community.

It claims to be the second-largest Stack Overflow community. So, if we ever get trapped in programming, other programmers or the Java community can help assist us and solve a problem.

XML :

XML (Extensible Markup Language) is commonly used in professional application development for various purposes, including:

1.User Interface Design:

XML is extensively used for defining the layout and structure of user interfaces in applications. In Android development, XML files are used to create layouts for activities, fragments, and custom views using elements such as TextView, ImageView, Button, etc. XML allows developers to specify the arrangement, appearance, and behavior of UI components in a declarative manner, separate from the application logic.

2. Data Exchange and Configuration:

XML is employed for data exchange and configuration purposes in professional applications. It provides a structured format for representing data in a human-readable and platform-independent manner. XML files are often used for storing configuration settings, defining data schemas, and exchanging data between different systems or components within an application. For instance, XML configuration files are commonly used in web applications for specifying database connections, security settings, and other runtime configurations.

3. Serialization and Persistence:

XML is utilized for serializing and persisting application data in a structured format. It enables developers to convert complex object graphs into XML documents that can be stored in files, databases, or transmitted over networks. XML serialization is commonly used in scenarios such as saving application state, storing user preferences, and exchanging data with external systems.

Additionally, XML-based technologies like XML Schema and XML-RPC provide standards for defining data structures and remote procedure calls, respectively, facilitating interoperability between different systems and platforms.

XML plays a crucial role in professional application development by providing a flexible, standardized, and widely adopted format for defining user interfaces, exchanging data, configuring applications, and persisting data, contributing to the efficiency, maintainability, and interoperability of modern software systems.

SQLite :

SQLite is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems.

Many programming languages have bindings to the SQLite library. It generally follows PostgreSQL syntax, but does not enforce type checking by default.This means that one can, for example, insert a string into a column defined as an integer.

- SQLite was designed to allow the program to be operated without installing a database management system or requiring a database administrator. Unlike client–server database management system.
- The SQLite engine has no standalone processes with which the application program communicates.
- Instead, a linker integrates the SQLite library statically or dynamically into an application program which uses SQLite's functionality through simple function calls, reducing latency in database operations; for simple queries with little concurrency, SQLite performance profits from avoiding the overhead of inter-process communication.
- SQLite stores the whole database (definitions, tables, indices, and the data itself) as a single cross-platform file on a host machine, allowing several processes or threads to access the same database concurrently.
- It implements this simple design by locking the database file during writing. Write access may fail with an error code, or it can be retried until a configurable timeout expires.
- SQLite read operations can be multitasked, though due to the serverless design, writes can only be performed sequentially

Why SQLite ?

1) SQLite does not require a separate server process or system to operate (serverless).
2) SQLite comes with zero-configuration, which means no setup or administration.
3) A complete SQLite database is stored in a single cross-platform disk file.
4) SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
5) SQLite is self-contained, which means no external dependencies.
6) SQLite supports most of the query language features found in SQL92 (SQL2) standard.
7) SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT)

KOTLIN

Kotlin is a modern programming language primarily used for Android app development, offering several advantages that make it popular in application development:

1. Conciseness and Readability:

Kotlin is designed to be concise, reducing the amount of boilerplate code required compared to Java.
Its expressive syntax enhances code readability, making it easier for developers to understand and maintain their applications.

2. Interoperability with Java:

Kotlin seamlessly interoperates with Java, allowing developers to leverage existing Java libraries, frameworks, and tools in their Kotlin projects.

This interoperability enables a smooth transition for developers migrating from Java to Kotlin and facilitates the adoption of Kotlin in existing codebases.

2. Safety and Null Safety:

Kotlin provides built-in null safety features, reducing the risk of Null Pointer Exceptions (NPEs) that are common in Java code.

Through its type system and nullable types, Kotlin ensures that nullability is explicitly handled, improving the reliability and stability of applications.

CRADLE :

1. Cradle is a build automation tool used primarily for Java projects, providing a convenient way to manage dependencies, compile code, run tests, and package applications.

2. Its primary use in application development revolves around simplifying the build process and managing project dependencies. With Cradle, developers can define project configurations, dependencies, and tasks in a build script (usually

written in Groovy or Kotlin DSL). This script specifies how the project should be built, including dependencies resolution, compilation, testing, and packaging.

3. Additionally, Cradle integrates seamlessly with popular IDEs like IntelliJ IDEA and Android Studio, allowing developers to easily import Cradle-based projects and leverage its build capabilities within their development environment.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

- The existing system does not have a proper channel of communication between the Farmer and the Consumer or the end user. Sometimes the farmers are cheated by the mediators and paid very less.

- There was always an Intermediate between the farmer and the end user, so the farmer will not a get a good pay for what he has cultivated.

- The amount estimated by the insurance company will not be as much accurate as our proposed system.

2.2 PROPOSED SYSTEM

In-order to avoid the limitation in the existing system is being developed.

- The current system that is going to be built overcomes all the difficulties given above.

- The Farmers are in direct contact with the consumer, so there is no need of the intermediate.

- The farmer will sell his product at a reasonable price, farmer would be able to know to whom he is selling his products and vice versa for the consumers too

- This would be possible through the facility added to this system for the communication Gmail or the SMS and the location sharing of the farmers land through GPS.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

- User friendly interface
- High Efficiency
- Fast access
- More Safety Utilization

- Cost Savings

3. SYSTEM DESIGN

3.1 FILE DESIGN

System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step inmoving from the problem domain to the solution domain. The design of the system is the critical aspect that affects the quality of the application. System design is also called top-level design. The design phase translates the logical aspects of the systeminto physical aspects of the system.

3.2 INPUT DESIGN

The data, which is input to a computer – based information system, must be correct. If data is carelessly input and errors enter the system, it will lead to incorrectresults whose consequences will be expensive and embarrassing to the designer. In data processing, the data entry operator often makes errors. This can be controlled byinput design by using menu, interactive dialogue, consistent format etc.

In this system the users are provided with user friendly pages to give the inputand if the user gives any wrong input validations are done and message boxes are provided in the necessary places. The message specified in the message box is specified in a polite and in an informative manner.
System is interactive dialogue, which simplifies the data entry or access, instead of remembering what to enter. User can choose from a list of options and type it in the cursor position. This will reduce the number of corrections while entering the data.

3.3 DATABASE DESIGN

The database design involves creation of tables that are represented in physicaldatabase as stored files. They have their own existence. Each table constitute of rowsand columns where each row can be viewed as record that consists of related information and column can be viewed as field of data of same type. The table is alsodesigned with some position can have a null value.

The database design of project is designed in such a way values are kept without redundancy and with normalized format. Refer the appendix for screen shots of database design.

3.4 OUTPUT DESIGN

The proposed system is a web oriented system and hence it does not provide any reports. The output results are viewed in the web pages itself. Outputs from the computer system are required primarily to communicate the result of processing to users. They are also used to override a permanent copy of the results for later consultation. The output reports and input documents should be documented in termsof data content .

4. SYSTEM TESTING AND IMPLEMENTATION

4.1 TESTING

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed systemworks according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a highprobability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test caseis a set of data that the system will process as an input. However the data are createdwith the intent of determining whether the system will process them correctly withoutany errors to produce the required output.

Types of Testing

- Unit Testing
- Integration Testing
- Output Testing
- User acceptance Testing
- Performance Testing
- Output Testing

Unit Testing

All modules were tested and individually as soon as they were completed and were checked for their correct functionality.

Integration Testing

The entire project was split into small program; each of these single programs gives a frame as an output. These programs were tested individually; at last all theseprograms where combined together by creating another program where all these constructors were used. It give a lot of problem by not functioning is an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. When the frames where given for the test, the end user gave suggestion. Based on their suggestions theframes where modified and put into practice.

Validation Testing

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., Validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

Output Testing

After performing the validation testing the next step is output testing of the proposedsystem. Since the system cannot be useful if it does not produce the required output.Asking the user about the format in which the system is required tests the output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is on screen and another one is printed format. The output format on the screen is found to be corrected as the format was designed in the system phase according to the user needs. And for the hardcopy the output comes according to the specifications requested by the user.

White box testing

White box testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of theitem being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential.
Whitebox testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, islike a white/transparent box; inside which one clearly sees.

Black box testing

Black box testing, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being testedis not known to the tester. These tests can be functional or non-functional, though usually functional.
This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see.

This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Definition by **ISTQB**

- **Black box testing:** Testing, either functional or non-functional, withoutreference to the internal structure of the component or system.
- **Black box test design technique:** Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non- functional, of a component or system without reference to its internal structure.

Acceptance testing

This testing is done to verify the readiness of the system for the implementation.Acceptance testing begins when the system is complete. Its purpose is to provide theend user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. Tools to specialimportance during acceptance testing include:

Test coverage Analyzer

Records the control paths followed for each test case.

Timing Analyzer

Also called a profiler, reports the time spent in various regions of the code are areasto concentrate on to improve system performance.

4.2 SYSTEM IMPLEMENTATION

The medical management System begins with the following involves variousactivities performed together. These are the System Development Life Cycle

i. Recognition of need

It is the first stage of information system development cycle. The preliminary investigation must define the scope of the project and the perceived constrains, opportunities and directives that triggered the project. As for Clinical Management System, I collected the system requirements throughquestionnaires and interviewing student and the staff and the problem they facewhen they visit the universities Clinic. I happen to find the following:

The preliminary investigation include the following tasks:

- Listing problems, opportunities and directives.
- Assess project worth.
- Plan the project.
- Present the project and plan.

ii.Feasibility study

The goal of a feasibility study is to evaluate alternative system and to purpose the most feasible and desirable system for development.

It consist of the following:

- Statement of the problem
- Summarizing of findings and recommendations
- Details of findings
- Recommendations and conclusions

I addressed five types of feasibility study in my research, they include the following.

1. Operational Feasibility

The system is operationally feasible.

1. Time Feasibility

Being a small system and given the period of three months of development, it is timefeasible.

2. Economic Feasibility:

A network-based system requires a lot of equipment such as cables, hubs etc. This requires a lot of initial capital to install the network. On the other hand, it allows sharing of resources and information and centralized administration hence cheaper.

3. Technical Feasibility

Since it is not a complex system, we have the technical feasibility of developing thesystem.

4. Time Feasibility

The system is a small one and hence the time frame of three months allocated for development is enough hence there is time feasibility.

From the above we choose to use a network based database system because as compared to the other strategies, it more feasible. It will contain an interface that is distributed in the network and is connected to a central data-base.

Feasibility study involve cost/benefit analysis. In the process , the cost and benefits are estimated with greater accuracy. If cost and benefit should bequantified to make a good system that is affordable.

iii Analysis

Analysis starts with systems request that describes the problems or desired changes in the system. It identifies the nature and scope of the business opportunity and problem by performing a feasibility study

iv Design

The Design phase creates a blueprint for the new system that will satisfy alldocumented requirements. It identifies all necessary outputs, inputs, interfaces and processes. Designs internal and external controls that will ensure:

- Reliability
- Security
- Maintainability
- Accuracy

The design is documented in the systems design specification and presented to the management and users for their review and approval. The involvement of Management and users is to avoid any misunderstanding about what the system willdo, how it will do it and how much it will cost.

Implementation

In the implementation phase, the new system is constructed by the programeers and designers and finally given to the final user.After implementation data is converted into system files,users are trained, and the actual transition to the new system is undertaken.

A Systems Evaluation is later done to determine If the systems operates properly andif the cost of the system and benefits are within expectations

Post implementation and maintenance

During this phase the IT department and staff maintains (corrects the errors and adapt to changes in the environment) and enhances the system. Enhancements provide a maximized return on IT investments

5. CONCLUSION

I believe I have done enough research on the Project and am ready to start and complete the project over the period specified and also make the Output.

With the help of this project,

- This project as described in detail, the application developed for the easy or direct selling of commodities using this android portal.
- The application enables the organization to carry out all the report effectively after the implementation.
- When all the suggestions forwarded in the software proposal have been successfully completed.

Thank you in advance for your consideration.

FUTURE ENHANCEMENT

- This project has been developed keeping in mind all the given possible conditions to overcome the disadvantages of existing system.
- This application is implemented successfully.
- After implementation of the application the user may require some changes to be made with the project.
- Whenever changes are made it will not affect the performance or efficiency of the existing system. Several navigations can be provided for easy access of data.

REFERENCES

BOOK REFERENCES

1)Wei-Meng Lee, “Beginning Android 4 Application Development”, Wiley India Pvt., Ltd., 2012

2)J.F. DiMarzio, “Android: A Programmers Guide”, First Edition, Tata McGraw-Hill, 2010.

3)Jason Morris, “Android User Interface Development”, Packt Publishing, 2011.

4) Jay A kreibich, “Using SQLite”, O’Reily Media,2010.

WEBSITES

- http://www.androiddeveloper.com/
- http://thenewboston.org/tutorials.php
- http://www.tutorialspoint.com//

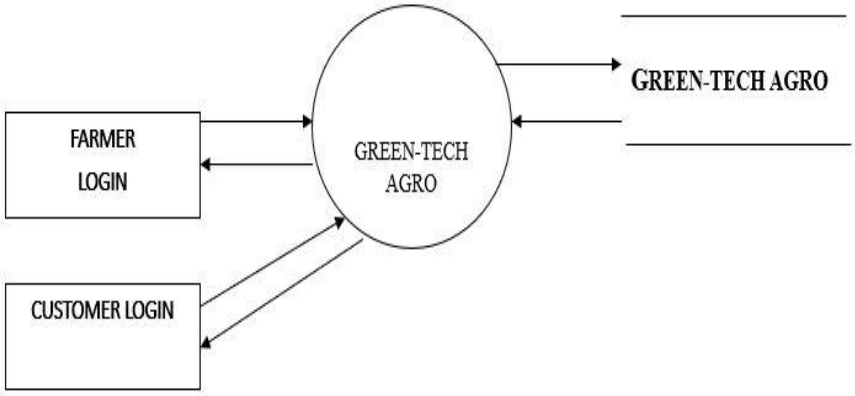
OTHER REFERENCES

https://www.scribd.com/document/38823393/Green-Tech-Industries-India-Pvt-Ltd-EXE-SUM-EnG

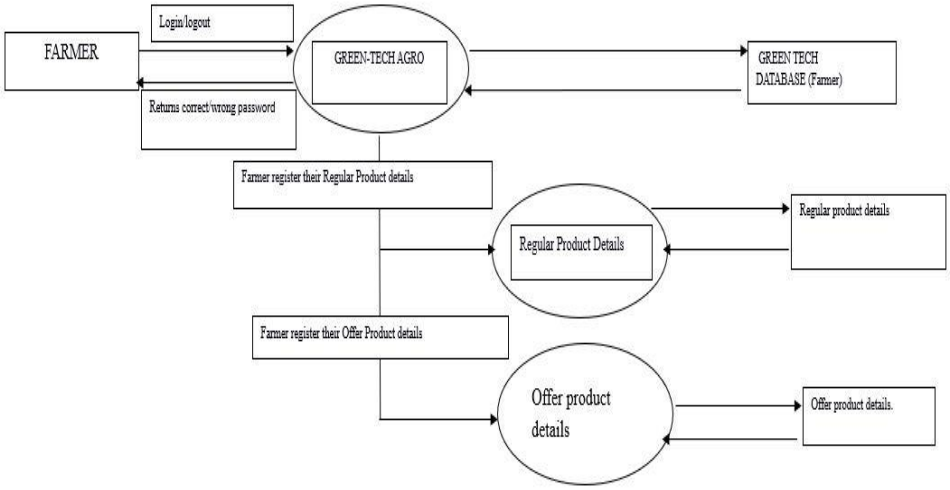
https://www.researchgate.net/publication/23516869_Green_Technologies_for_aMore_Sustainable_Agriculture

https://www.greentech.at/wp-content/uploads/2021/03/Green_Tech_Guide_2017.pdf

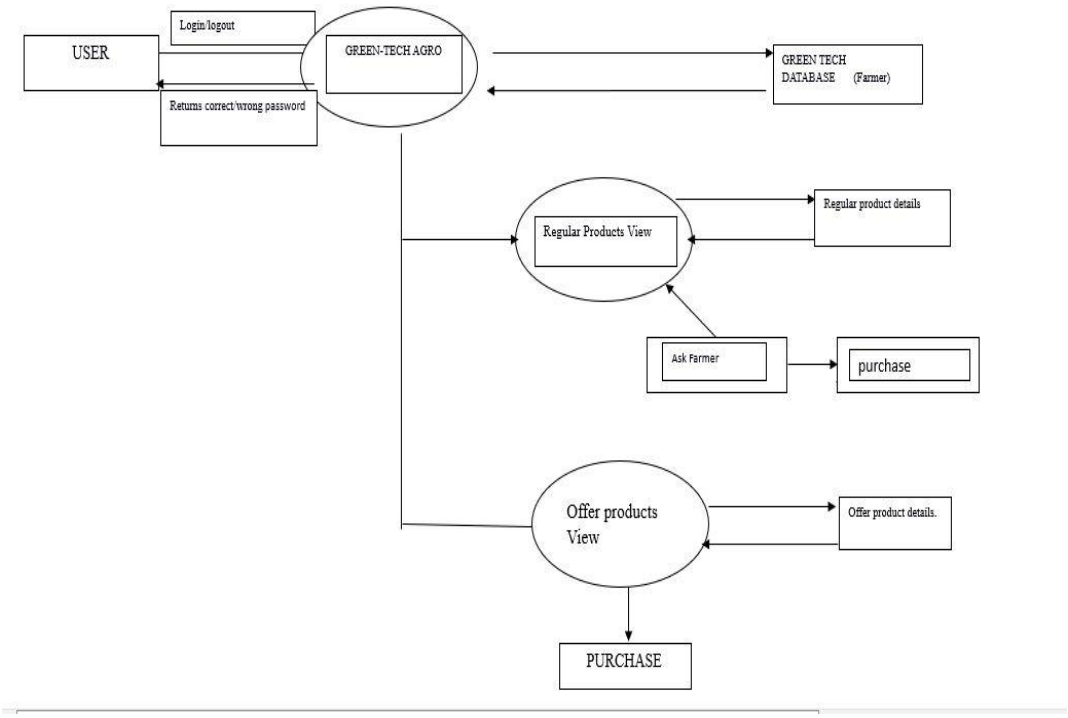
FROM DESIGN



LEVEL 1 (FARMER) :



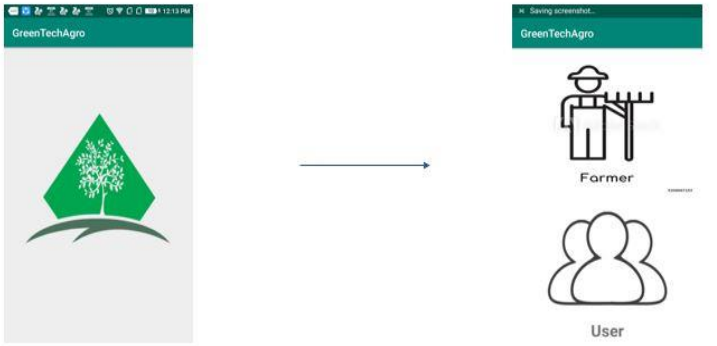
LEVEL 2 (USER) :


USER
Login/logout
Returns correct/wrong password
GREEN-TECH AGRO
GREEN TECH
DATABASE (Farmer)
Regular Products View
Regular product details
Ask Farmer
purchase
Offer products
View
Offer product details.
PURCHASE

OUTPUT SCREENSHOT

SCREENSHOT 1:

INPUT/OUTPUT DESIGN



SPLASH SCREEN DASHBOARD SCREEN

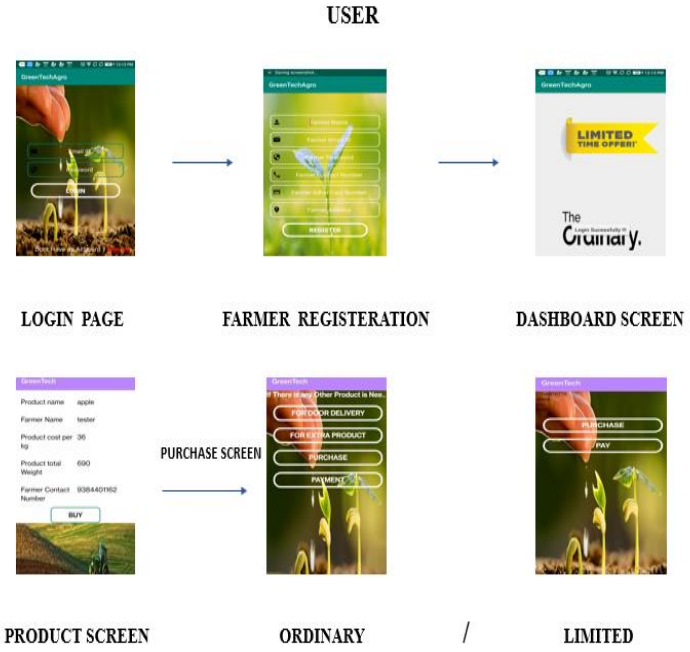
SCREENSHOT 2:

FARMER

LOGIN PAGE FARMER REGISTERATION DASHBOARD SCREEN

ADD ORDINARY / ADD LIMITED REGISTERED SCREEN

SCREENSHOT-3



CODING 1 :

```
package com.example.greenagri;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.drawerlayout.widget.DrawerLayout;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
public class FarmerDashboardActivity extends AppCompatActivity {
    ImageButton offer,ordinary;
    public DrawerLayout drawerLayout;
    public ActionBarDrawerToggle actionBarDrawerToggle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_farmer_dashboard);
offer =  findViewById(R.id.offerproduct);
ordinary =  findViewById(R.id.ordinarproduct);
drawerLayout = findViewById(R.id.my_drawer_layout);
actionBarDrawerToggle = new ActionBarDrawerToggle(this, drawerLayout, R.string.nav_open, R.string.nav_close);
// pass the Open and Close toggle for the drawer layout listener
// to toggle the button
drawerLayout.addDrawerListener(actionBarDrawerToggle);
actionBarDrawerToggle.syncState();
// to make the Navigation drawer icon always appear on the action bar
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
offer.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Intent in = new Intent(FarmerDashboardActivity.this,OfferActivity.class);
      startActivity(in);
   }
});
ordinary.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Intent in = new Intent(FarmerDashboardActivity.this,FarmerProductActivity.class);
      startActivity(in);
   }
});}
public boolean onOptionsItemSelected(MenuItem item) {
   // Switching on the item id of the menu item
   switch (item.getItemId()) {
      case R.id.nav_account:
         startActivity(new Intent(getApplicationContext(),LoginScreen.class));
         return true;
      case R.id.nav_settings:
         startActivity(new Intent(getApplicationContext(),SettingActivity.class));
         return true;
      case R.id.nav_logout:
         startActivity(new Intent(getApplicationContext(),FarmerLoginActivity.class));
         return true;
      default:
```

```
            return super.onOptionsItemSelected(item);
        }
    }
}
```

CODING 2:

```
package com.example.greenagri;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.List;
public class FarmerofferActivity extends AppCompatActivity {
    List<OfferProduct_View> employeeList;
    SQLiteDatabase mDatabase;
    ListView listViewEmployees;
    OfferProductAdapter_View adapter;
    LoginDataBaseAdapter loginDataBaseAdapter;
    TextView name, mail, num;
    String names,mails,nums;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_farmeroffer);
        loginDataBaseAdapter = new LoginDataBaseAdapter(this);
        loginDataBaseAdapter = loginDataBaseAdapter.open();
        name = (TextView) findViewById(R.id.onameview);
        mail = (TextView) findViewById(R.id.oemailview);
        num = (TextView) findViewById(R.id.omobview);
        listViewEmployees = (ListView) findViewById(R.id.olist);
        employeeList = new ArrayList<>();
        //opening the database
        mDatabase = openOrCreateDatabase(OfferActivity.DATABASE_NAME, MODE_PRIVATE, null);
        //this method will display the employees in the list
```

```
        showEmployeesFromDatabase();
    }
    private void showEmployeesFromDatabase() {
        Cursor cursorEmployees = mDatabase.rawQuery("SELECT * FROM OfferProduct", null);
        //if the cursor has some data
        if (cursorEmployees.moveToFirst()) {
            //looping through all the records
            do {
                //pushing each record in the employee list
                employeeList.add(new OfferProduct_View(
                        cursorEmployees.getString(0),
                        cursorEmployees.getString(1),
                        cursorEmployees.getString(2),
                        cursorEmployees.getString(3),
                        cursorEmployees.getString(4)
                ));
            } while (cursorEmployees.moveToNext());
        }
        cursorEmployees.close();
        //creting the adapter object
        adapter = new OfferProductAdapter_View(this, R.layout.offer_farmer_view_list, employeeList, mDatabase);
        //adding the adapter to listview
        listViewEmployees.setAdapter(adapter);
    }
}
```

CODING 3 :

```
package com.example.greenagri;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class FarmerLoginActivity extends AppCompatActivity {
```

```
EditText mail, pass;
Button login;
TextView signup;
LoginDataBaseAdapter loginDataBaseAdapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_farmer_login);
   loginDataBaseAdapter = new LoginDataBaseAdapter(this);
   loginDataBaseAdapter = loginDataBaseAdapter.open();
   mail = (EditText) findViewById(R.id.Femaillogin);
   pass = (EditText) findViewById(R.id.Fpasslogin);
   login = (Button) findViewById(R.id.Floginbt);
   signup = (TextView) findViewById(R.id.Fsignup);
   signup.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
         Intent in = new Intent(FarmerLoginActivity.this, FarmerRegister.class);
         startActivity(in);
      }
   });

   login.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
         String email = mail.getText().toString();
         String pas = pass.getText().toString();
         String temp = loginDataBaseAdapter.getSinlgeEntryF(email);
         if (temp.equals(pas)) {
            Intent in = new Intent(FarmerLoginActivity.this,Farmer_selction.class);
            startActivity(in);
            Toast.makeText(getApplicationContext(), "Login Sucessfully !!!", Toast.LENGTH_LONG).show();
         } else {
            Toast.makeText(getApplicationContext(), "Login Failed !!!", Toast.LENGTH_LONG).show();
            mail.setText("");
            pass.setText("");
         }
      }
   });}}
```

CODING 4:

```
package com.example.greenagri;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
public class DataBaseHelper extends SQLiteOpenHelper {
    public DataBaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }
    // Called when no database exists in disk and the helper class needs
    // to create a new one.
    @Override
    public void onCreate(SQLiteDatabase _db)
    {
        _db.execSQL(LoginDataBaseAdapter.DATABASE_USER);
        _db.execSQL(LoginDataBaseAdapter.DATABASE_FARMER);
    }
    // Called when there is a database version mismatch meaning that the version
    // of the database on disk needs to be upgraded to the current version.
    @Override
    public void onUpgrade(SQLiteDatabase _db, int _oldVersion, int _newVersion) {
        // Log the version upgrade.
        Log.w("TaskDBAdapter", "Upgrading from version " + _oldVersion + " to " + _newVersion + ", which will
destroy all old data");
        // Upgrade the existing database to conform to the new version. Multiple
        // previous versions can be handled by comparing _oldVersion and _newVersion
        // values.
        // The simplest case is to drop the old table and create a new one.
        _db.execSQL("DROP TABLE IF EXISTS " + "TEMPLATE");
        // Create a new one.
        onCreate(_db);
    }
}
```