

**AGRO BAZAR**  
**Mini Project Report**

Submitted by  
**ANAGHA M U**

**Reg.No : FIT20MCA-2016**

*Submitted in partial fulfillment of the requirements for the award of  
the degree of*

*Master of Computer Applications  
Of  
A P J Abdul Kalam Technological University*



**Focus on Excellence**

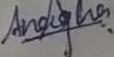
**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®  
ANGAMALY-683577, ERNAKULAM(DIST)  
MARCH 2022**

## **DECLARATION**

I, ANAGHA M U hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (FISAT), Angamaly in partial fulfillment of the award of the degree of Master of Computer Applications is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

signature 

Place: Angamaly

name: Anagha M U

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®  
ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

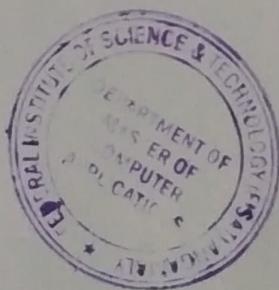


CERTIFICATE

This is to certify that the project report titled "AGRO BAZAR" submitted by ANAGHA M U [Reg.No : FIT20MCA-2016] towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

*[Signature]*  
Project Guide Dr. Shreya M V

*[Signature]*  
Head of the Department  
Dr. Deyya May Mathew



## **ACKNOWLEDGEMENT**

I hereby deeply grateful to **Dr.Manoj George** , Principal, FISAT, Angamaly for providing me with adequate facilities, way and means by which I were able to complete my mini project work and I express my sincere thanks to **Dr. C. Sheela** ,Vice Principal FISAT, Angamaly.

my sincere thanks to **Dr.Deepa Mary Mathwes**, Head of the department of Master of computer application, FISAT, who had been a source of inspiration. I express heartiest thanks to **Dr.Shidha M V** my project guide for the encouragement and valuable suggestion . I express my heartiest gratitude to my scrum master **Ms.joice T** and the faculty members in my department for the constant encouragement and never ending support throughout the project.I would also like to express my sincere gratitude to the lab faculty members for their guidance

Finally I express thanks to all my friends who gave me wealth of suggestion for successful completion of this project.

## **ABSTRACT**

Agro-bazar software is user-friendly web application which is used to connect farmers with customers. I understand that the customers who are compelled to buy the toxic vegetables and fruits by paying high price. Agro-bazar brings a solution for this. With the help of Agro-bazar, farmers can directly sell their products to the customers. Pure, fresh non-toxic vegetables, fruits and food products are delivered to customers. Customers can get these items at a lower price as we eliminate the third parties in between.

Farmers can add their products to agro-bazar portal were it can be viewed and purchased by a large group of users. By purchasing items directly from farmers have mainly two advantages. Customers can get pure and non-toxic items. Secondly by avoiding the intermediate people between farmers and customers, the items can be purchased at a reasonable price.

Agro-bazar also provide option to collect items from farmers to a central hub and distribute them to the needed customers. The customers can also avail home delivery facilities offered by agro-bazar. The customers can also provide feedback for the purchased product. If the customers provide negative feedback about an item of a farmer, the admin controller have the provision to suspend that item / farmer from the system.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>8</b>
<b>2</b>	<b>SYSTEM ANALYSIS</b>	<b>10</b>
2.1	EXISTING SYSTEM . . . . .	10
2.2	PROPOSED SYSTEM . . . . .	11
2.3	SYSTEM SPECIFICATION . . . . .	12
2.4	SOFTWARE SPECIFICATION . . . . .	12
2.5	HARDWARE SPECIFICATION . . . . .	13
<b>3</b>	<b>IMPLEMENTATION</b>	<b>15</b>
3.1	TOOLS AND PLATFORM . . . . .	15
<b>4</b>	<b>SYSTEM DESIGN</b>	<b>19</b>
4.1	INPUT DESIGN . . . . .	20
4.2	OUTPUT DESIGN . . . . .	21
4.3	DATABASE DESIGN . . . . .	23
4.4	DATA FLOW DIAGRAM . . . . .	23
4.5	TABLE DESIGN . . . . .	28
<b>5</b>	<b>SYSTEM DEVELOPMENT</b>	<b>33</b>
5.1	MODULE DESCRIPTION . . . . .	33
5.1.1	Registration and Login . . . . .	33
5.1.2	Buy and Sell . . . . .	33

5.1.3	Delivery System . . . . .	33
<b>6</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>34</b>
6.1	TESTING . . . . .	34
6.1.1	UNIT TESTING . . . . .	35
6.1.2	INTEGRATION TESTING . . . . .	36
6.1.3	USER ACCEPTANCE TESTING . . . . .	36
6.1.4	OUTPUT TESTING . . . . .	36
6.2	VALIDATION CHECK . . . . .	37
<b>7</b>	<b>SYSTEM MAINTENANCE AND FUTURE ENHANCEMENTS</b>	<b>38</b>
7.1	SYSTEM MAINTENANCE . . . . .	38
7.2	FUTURE ENHANCEMENTS . . . . .	39
<b>8</b>	<b>CODING</b>	<b>40</b>
8.1	SOURCE CODE . . . . .	40
8.2	SCREENSHOTS . . . . .	48
8.2.1	ADMIN PAGE . . . . .	48
8.2.2	LOGIN PAGE . . . . .	52
8.2.3	CUSTOMER PAGE . . . . .	53
8.2.4	FARMER PAGE . . . . .	54
8.2.5	DELIVERY BOY PAGE . . . . .	55
<b>9</b>	<b>CONCLUSION</b>	<b>56</b>
9.0.1	BIBLIOGRAPHY . . . . .	57

# **Chapter 1**

## **INTRODUCTION**

The first step in the system study includes analysis of the system. System analysis involves studying the way an organization currently receives and process data to produce information with the goal of determining how to make it work better. System analysis includes both a preliminary and a detailed stage. During preliminary analysis the analysis takes a quick look at what is needed and whether it benefits the perceived want. Detailed analysis includes an in depth look at what is wanted and contains more refined cost and benefits studies. The preliminary analysis begins when someone perceives a problem, modifications to existing, repairs to an existing system or demands an entirely new system. The analyst summarizes the gained modifications, including personal requirements and potential benefits of the new system in formal report called the preliminary report. Detailed analysis expands the preliminary efforts to include the complete analysis of all possible alternative solutions to the problem and complete expansion of what appears to be the most practical solution.

The system study is the process of gathering and interpreting facts, using this information for further studies on the system. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decisions about the strategies to be followed for effective system study and analysis can be taken. The system study also identifies the method

collection to be followed. The system study conducted an initial picture about the system working was got. From the information got from the study, the data collection methods are identified. Even in the first investigation itself drawbacks of the existing system could be identified.

Analysis involves the requirement determination and specifications. Basically, it involves establishing for all the system elements and then mapping these requirements to the software form. The analysis is intended to capture and describe all the requirements of the system and to make a model that defines a key domain class in the system. The purpose is to provide an understanding and to enable a communication about the system between the developers and the people establishing the requirements. Therefore, the analysis is typically terms of code or programs during this phase; it is the first step towards really understanding the requirements.

# **Chapter 2**

## **SYSTEM ANALYSIS**

### **2.1 EXISTING SYSTEM**

The current system of shopping is to visit the shop manually and from the available agricultural products choose the items customer wants and buying the item by payment of the price of the item. This method needs more man power to manage the shop. Customers have to walk in to the shop to buy the products, the product list is limited and description of the product is also limited. It is hard to find the required products.

- It is fewer users friendly.
- User must go to shop and select product.
- Involves a lot of human efforts.
- There is no support for local/small farmers.
- Non-toxic products
- Time consuming
- Description of the product is limited.
- Availability of seasonal fruits and vegetables

## 2.2 PROPOSED SYSTEM

The proposed system, customer need not go to the shop for buying the agricultural product he can order the product he wishes to buy through the application. The system also recommends a home delivery system for the purchased products.

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

### FEATURES:

The proposed system is designed to eliminate all the drawbacks of the existing system. The system has been responsible for maintaining information about users,

- Farmer registration
- Customer registration
- Delivery boy registration
- View agricultural products
- Add products to system
- Add products to cart
- Checkout products and payment
- Add farmer and delivery boy
- Home delivery

## 2.3 SYSTEM SPECIFICATION

A software requirement specification (SRS), a requirements specification for a software system, is a complete description of the behaviour of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition, it also contains non-functional requirements. Nonfunctional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed.

## 2.4 SOFTWARE SPECIFICATION

The selection of software is very important in the existence and proper working of any software. When selecting software, the size and capacity requirements are also important. Below are some of the software that is required for the system.

Operating System	64-bit Microsoft® Windows® 8/10/11
Programming Language	Java, Android
RDBMS	MYSQL
Web Server	Apache Tomcat & Glassfish Server
Scripting language	JSP

## 2.5 HARDWARE SPECIFICATION

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. I require much different software to make the application which is in making to work efficiently. It is very important to select the appropriate software so that the software works properly.

Main Processor	Intel core i3 or above
RAM	8 GB or Above
Keyboard	Standard 108 keys
Mouse	3D Optical mouse
Monitor	15" Standard
Hard disk	10 GB of available disk space minimum or Above

# **Chapter 3**

## **IMPLEMENTATION**

The implementation phase of the Agro Bazar Android application is connects farmers with customers for buying farming products. Thus it translates design specification into source code. The user(Admin as per Agro Bazar) tests the developed system and changes are made according to their needs. In this system, it has been successfully implemented. Before implementation, several tests have been conducted to ensure that no errors are encountered during the operation. This project's implementation is done as there are 4 users i.e., Admin, Farmer, Delivery person, Customer. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right.

### **3.1 TOOLS AND PLATFORM**

#### **FRONT END:ANDROID**

Android is a Linux based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers. Android is open source and Google releases the code under the Apache License. This open-

source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wire- less carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications that extend the functionality of devices, written primarily in a customized version of the Java programming language. Android is an open-source mobile operating system that combines and builds up on parts of many different open-source projects.

Android's kernel is based on the Linux kernel and has further architecture changes by Google outside the typical Linux kernel development cycle. Android does not have a native Window System nor does it support the full set of standard GNU libraries, and this makes it difficult to port existing Linux applications or libraries to Android. Android's user interface is based on direct manipulation using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on screen objects. Android devices boot to the home screen, the primary navigation and information point on the device, which is similar to the desktop found on PCs. Android home screens are typically made up of applications and widgets; applications launch the associated app, whereas widgets display live, auto updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen.

Graphical user interfaces  
frameworks

Multimedia

Database

Networking

## ABOUT JAVA

Java is a general purpose computer programming language that is concurrent, class-based, object oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers quote;write once, run anywherequot; (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling, a Canadian, at Sun Micro systems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Micro systems<sup>39</sup>; Java platform. The language derives much of its original features from Small talk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plugin for applets).

The latest version is Java 11, released on September 25, 2018. Java 11 is a currently supported long-term support (LTS) version (quote;Oracle Customers will receive Oracle Premier Supportquot;); Oracle released for the quote;legacyquot; Java 8 LTS the last quote;public updatequot;, which is free for commercial use, in January 2019. Oracle will still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) quote;highly recommend

that you uninstall older versions of Java; because of serious risks due to unresolved security issues. Since Java 9 is no longer supported, Oracle advises its users to immediately transition to Java 11. Oracle extended support for Java 6 ended in December 2018.

## **ABOUT MYSQL**

MySQL is a relational database management system (RDBMS) which is more than 11 million institutions. The program runs as a server providing multi-user access to a number of databases. MySQL is owned and sponsored by a single for profit firm, the Swedish company MySQL AB, now a subsidiary of Sun Microsystems, which holds the copyright to most of the code base. The project's source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements.

# **Chapter 4**

## **SYSTEM DESIGN**

System design is an interactive process through which requirement are transmitted to a “blue print” for constructing the software initial; the blue print depicts a holistic view of software that is design is represented at a high-level abstraction a level that can be directly traced to specific data, functional and behavioral requirement. System design is the solution to the creation of a new system. This is the important aspect made up of several steps. System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and output reports in a form applicable to the users.

The main objective of system design is to use the package easily by a computer operator. System design is the creative act of invention, developing new inputs, a database, off-line files, method, procedures and output for processing business to meet an organization objective. System design-built information gathered during the system analysis. As design interaction occur subsequent refinement leads to design representation at much lower level of abstraction. System design is a creative art of inventing and developing input, data bases, off line files, method and procedures, for processing data to get meaning full output that satisfy the organization objectives. Through the design phase consideration to the human factor, that is inputs to the users will have on the system.

## 4.1 INPUT DESIGN

Input design is the process of converting user-oriented input to a based format. Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. The goal of designing input data is to make data entry as easy, logical and free from errors. When we approach input data design; we design the data source documents that capture the data and then select the media used to enter them into computer.

User-friendly screen format can reduce the burden on end users, who are not highly proficient in computers. An important step in input design stage is a design of source document. Source document is the form in which the data can initially capture. The next step is the design of the document layout. In the layout organizes the document by placing information, where it will be noticed and establishes the appropriate sequence of items. User interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user oriented inputs into the computer based format. Input design is apart of overall system design, which requires very careful attention. If data going into the system is correct, then the processing and output will magnify these errors. Thus, the designer has a number of clear objectives in the different stages of input design

1. To produce a cost-effective method of input.
2. To achieve the highest possible level of accuracy.
3. To ensure that input is acceptable to and understand by the user.

In accurate input data is most common cause of data processing errors. If poor input design-particularly where operators must enter data from source documents permits bad data to enter a computer system, the outputs produced are of little value. The input design process was initiated in the study phase were, as a

part of the feasibility study:

1. Input data were found to be available for establishing and maintaining master and transaction files and for creating output records
2. The most suitable types of input media, for either off-line or on-line devices were selected after a study of alternative data capture techniques

The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. This data is validated where verity requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue forgiving the right messages for the user at the right time are also considered for development for this project.

## 4.2 OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship and helps user decision-making.

In the output design it is determine how the implementation is to be played for immediate need and also the hard copy output. A major form of input is a hard copy from the printer. Print-outs should be designed around the output requirement of the user. Printers, CRT screen display are the examples for providing computer based output. The output design associated with the system includes the various reports of the table generations and query executions. A quality output is one, which meets the requirements of end user and presents the information

clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need. It is the most important and direct source in formation to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision making.

The objective of the output design is to convey the information of all the past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users. Output design is one of the, most important features of the information system. The logical design of an information system is analogous to an engineering blue print of an automobile. It shows the major features and how they are related to one another. The outputs, inputs and databases are designed are in this phase. At the beginning of the output design various types of outputs such as external, internal, operational, and interactive and turnaround are defined. Then the format, content, location, frequency, volume and sequence of the outputs are specified. The content of the output must be defined in detail. The system analysis has two specific objectives at this stage.

- To interpret and communicate the results of the computer part of a system to the users in a form, which they can understand, and which meets their requirements.
- To communicate the output design specifications to programmers in a way in which it is unambiguous, comprehensive and capable of being translated into a programming language.

## 4.3 DATABASE DESIGN

The Database design is the process of producing a detailed data model of a database. The logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributes for each entry. The term database design can be used to describe many different parts of the design of an overall data base system. Principally, and most correctly, it can be thought to fast the logical design of the base data structures used to store the data. In the relational model, these are the tables and views. In an object data base, the entities and relationships map directly to object classes and named relationships. However, the term data base design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall data base applications within the database management system.

## 4.4 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) or a bubble chart is a graphical tool for structured analysis. It was De Marco in 1978 and Gene and Carson in 1979 who introduced DFD. DFD models a system transforms the data and creates, output data-flows which go by using external entities from which data flows to a process which to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in DFD. Bubbles represent the process. Named arrows indicate the dataflow. External entities are represented by rectangles and are outside the system such as vendors or customers with whom the system interacts. They either supply or consume data. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored

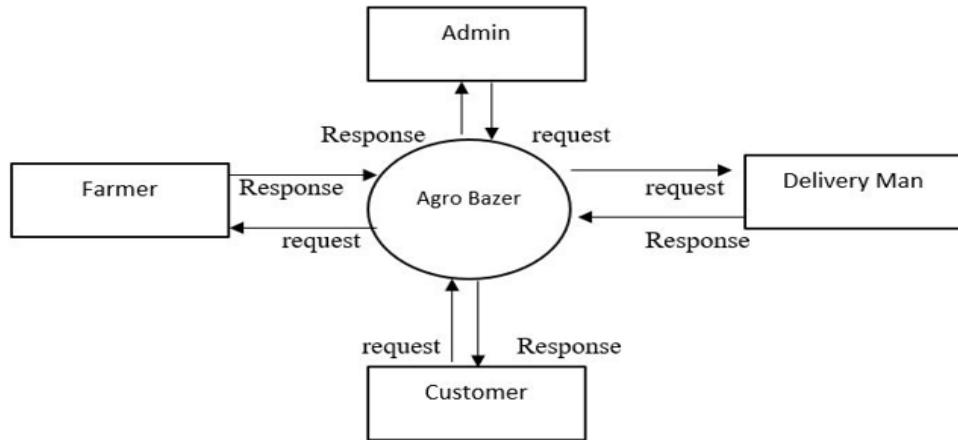
in a data store by a process in the system. Each component in a DFD is labelled in with a descriptive name. Process names are further identified with a number.

DFD can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one Data Flow Diagram can depict an entire system. Which gives the system overview? It is called Context Diagram of level 0 DFD. The context diagram can be further expanded. The successive expansion of the DFD from the context diagram that giving more details is known as levelling of DFD. Thus of top down approach is used, starting with an overview and then working out the details.

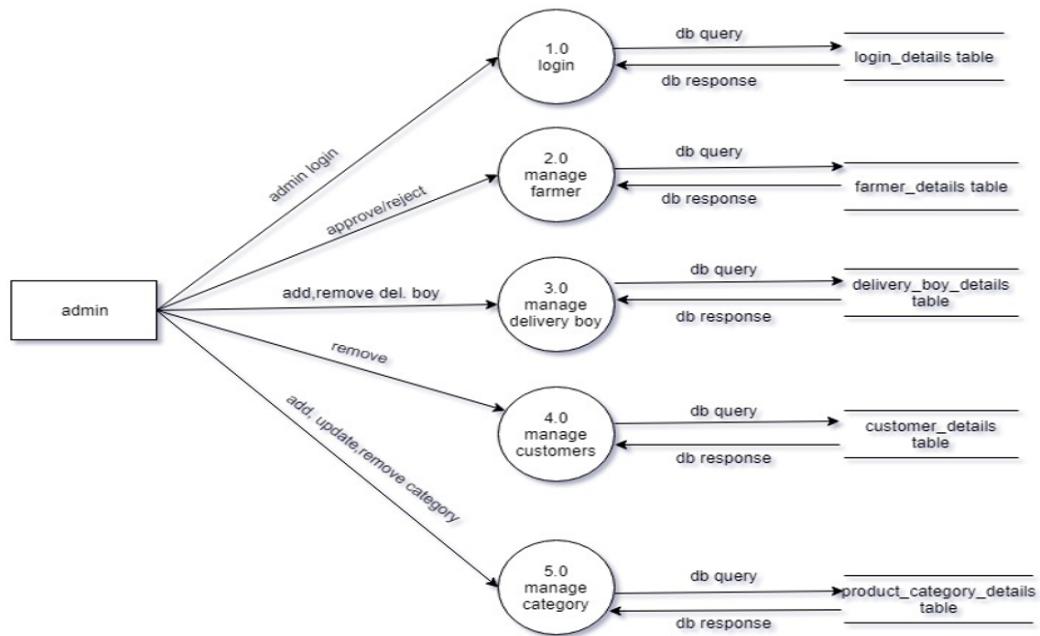
The main merit of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used, and where the result flow. The data flow diagram of Co-operative Bank Management System has been represented as a hierarchical DFD context level DFD was drawn first. Then the processes were decomposed into several elementary levels and are represented in the order of importance.

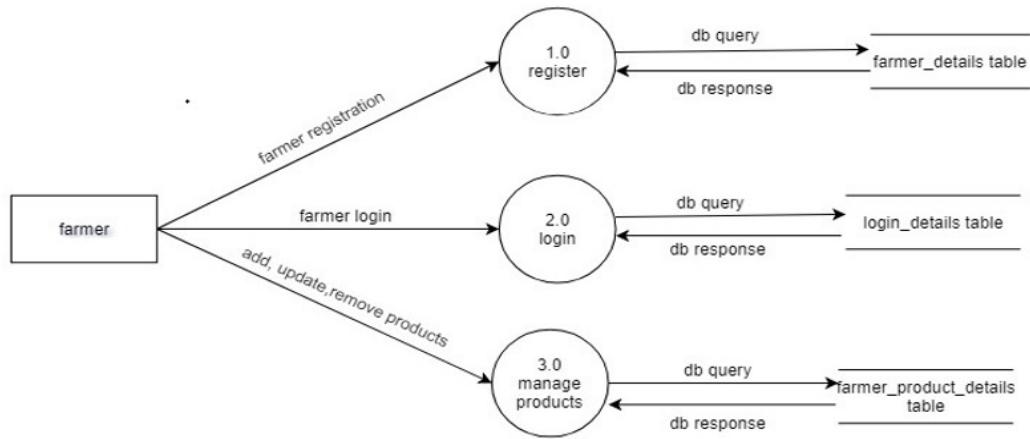
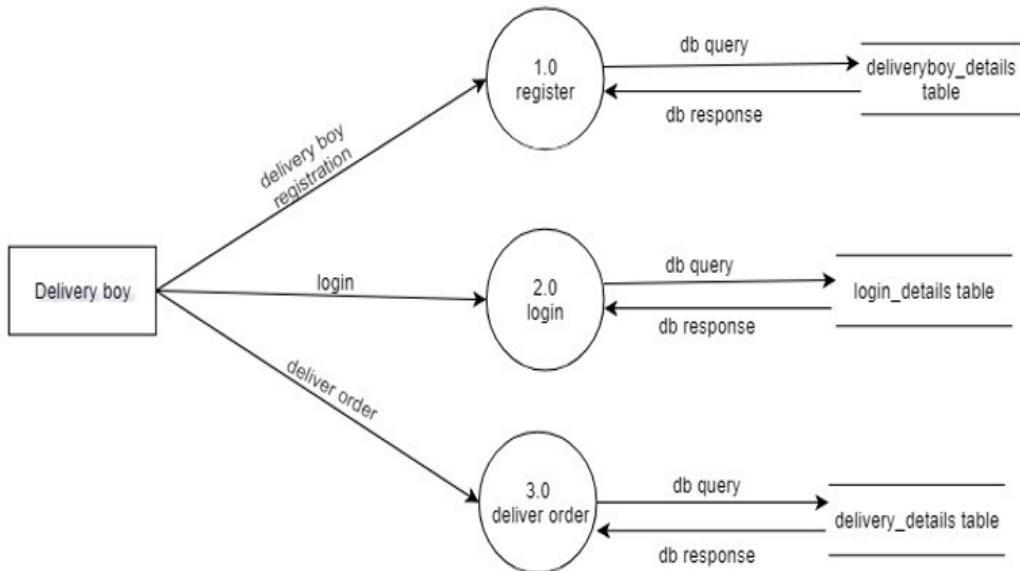
## DATA FLOW DIAGRAM

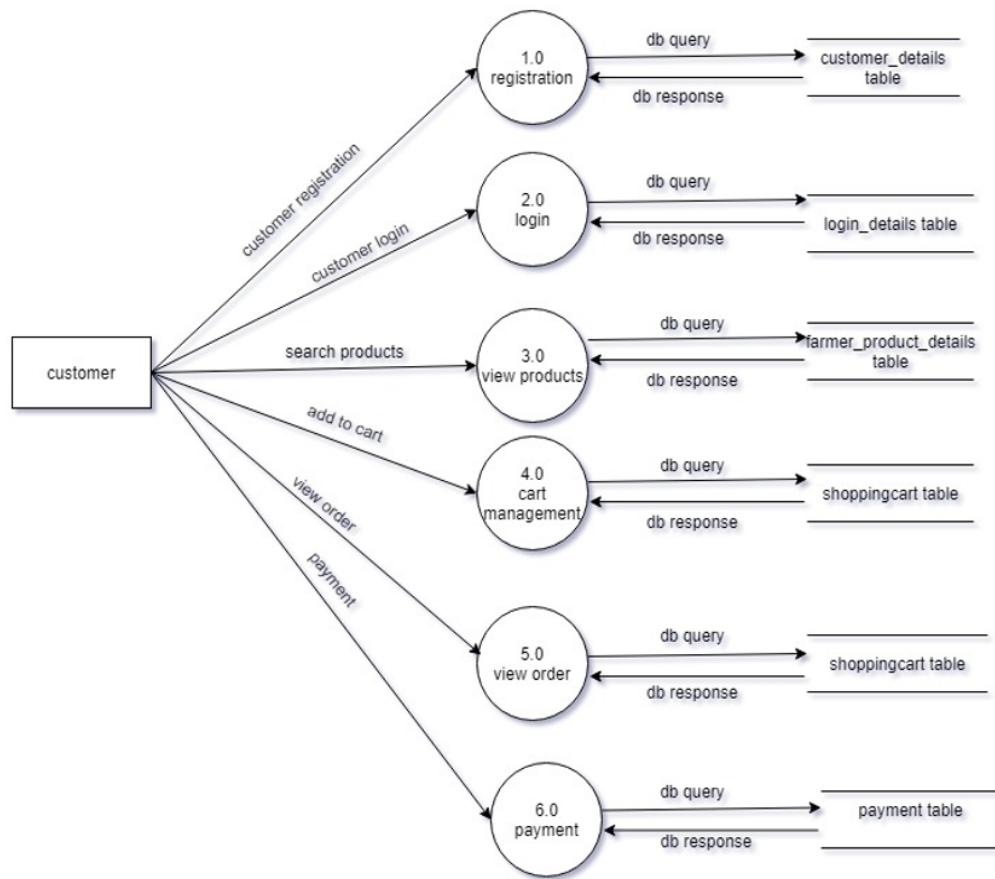
### Level -0



### Level-1 DFD for Admin



**Level-1 DFD for Farmer****Level-1 DFD for Delivery Boy**

**Level-1 DFD for Customer**

## 4.5 TABLE DESIGN

Table 4.1: Admin Details

Admi Details		
Column Name	Datatype	Description
Admin_id	Int	Id for admin
admin_email	Varchar	email id
admin_contact_no	Varchar	contact number
Admin_dob	date	date of birth
admin_address	Varchar	address

Table 4.2: Farmer Details

Farmer Details		
Column Name	Datatype	Description
Farmer_id	Int	Id for admin
farmer_fname	Varchar	first name
farmer_lname	Varchar	last name
farmer_email	Varchar	email id
farmer_contact_no	Varchar	contact number
farmer_gender	Varchar	Gender
Farmer_dob	date	date of birth
Farmer_address	Varchar	Address
Farmer_rating	Int	farmer rating

Table 4.3: Customer Details

Customer Details		
Column Name	Datatype	Description
Customer_id	Int	Id for customer
customer_fname	Varchar	first name
customer_lname	Varchar	last name
Customer_email	Varchar	email id
customer_contact_no	Varchar	contact number
customer address	Varchar	Address
customer rating	Int	customer rating

Table 4.4: Delivery Boy Details

DeliveryBoyDetails		
ColumnName	DataType	Description
delivery_boy_id	Int	Id for admin
delivery_boy_fname	Varchar	first name
delivery_boy_lname	Varchar	last name
delivery_boy_email	Varchar	email id
delivery_boy_contact_no	Varchar	contact number
delivery_boy_gender	Varchar	Gender
delivery_boy_address	Varchar	Address
delivery_boy_rating	Int	delivery boy rating

Table 4.5: Login Details

LoginDetails		
ColumnName	DataType	Description
user_id	Int	user id
user_type	Varchar	user type(admin,farmer,customer,delivery boy)
user_name	Varchar	user name
user_password	Varchar	password

Table 4.6: Product Category Details

ProductCategoryDetails		
ColumnName	DataType	Description
category_id	Int	id of product category
category_name	Varchar	category name

Table 4.7: Product Details

ProductDetails		
ColumnName	DataType	Description
product_id	Int	id of product
product_name	Varchar	product name
product_category_id	Int	product category id
product_description	Varchar	product description
product_image	Varchar	product image

Table 4.8: Farmer Product Details

<b>FarmerProductDetails</b>		
<b>ColumnName</b>	<b>DataType</b>	<b>Description</b>
farmer_id	Int	id of farmer
product_id	Int	id of product
product_price	Float	price of product
product_quantity	Varchar	quantity of product
product_description	Varchar	description of product
product_image	Varchar	image of product

Table 4.9: Shopping Cart

<b>ShoppingCart</b>		
<b>ColumnName</b>	<b>DataType</b>	<b>Description</b>
shopping_cart_id	Int	id of shopping cart
customer_id	Int	id of customer
created_time	timestamp	created time
modified_time	timestamp	modified time
order_status	Varchar	status of order
order_feedback	Varchar	feedback of order

Table 4.10: Cart Items

CartItems		
ColumnName	DataType	Description
cart_item_id	Int	id of cart item
product_id	Int	id of product
product_quantity	Float	quantity of product
shopping_cart_id	Int	id of shopping cart
created_time	timestamp	created time
modified_time	timestamp	modified time

Table 4.11: Delivery Details

DeliveryDetails		
ColumnName	DataType	Description
delivery_id	Int	id of order delivery
delivery_boy_id	Int	id of delivery boy
customer_id	Int	id of customer
shopping_cart_id	Int	id of shopping cart
delivery_status	varchar	status of the order delivery
delivery_time	timestamp	time of delivery

# **Chapter 5**

## **SYSTEM DEVELOPMENT**

### **5.1 MODULE DESCRIPTION**

#### **5.1.1 Registration and Login**

In our system, Admin should have full control of the system and include the activities manage farmer, manage delivery boy, manage products and manage delivery boy and managing all their registration and login activities.

#### **5.1.2 Buy and Sell**

In our system, Farmer are the sellers, where they can login and add products with details, update product and remove product. Farmers can view feedback from customers. The customers are the buyer of this task. Customer include the activities sign-up to system, view products add items to cart, purchase items using bank account details, send feedbacks.

#### **5.1.3 Delivery System**

In our system, Delivery boy include the activities, sign-up to system, sign-in and view orders deliver it to the customers

# **Chapter 6**

## **SYSTEM IMPLEMENTATION**

### **6.1 TESTING**

Testing is an important stage in the software development life cycle. System testing is a critical element of a software quality assurance and represents the ultimate review of specification, design and coding.

Importance of software testing and its implication with software quality cannot be over emphasized. Testing is one way developers can validate the quality of a software product and verify that it fully meets the specification. During testing, the system is tested with a set of cases and checked whether the input of the program is performing as it is expected. The system tested and reviewed to ensure that the entire user requirement has being satisfied.

Testing was done throughout the system development at various stages since it is always a good practice to test the system at many different levels at various intervals that is sub systems, program modules as work progress and finally the system as a whole. If this is not done, then the poorly tested system can fail after installation. Testing is a very tedious and time-consuming job. For a test to be successful the tester should try and make the program file. Each test is designed with the intention of finding errors in the way system will process it. Though testing of a program doesn't guarantee the reliability of the system, it

is done to assure that the system runs errors free. The Testing process begins by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The Process verifies that the application meets the requirements specified in the system requirements document and is bug free.

At the End of each testing day, the summary of completed and failed tests is prepared. And the Application is redeveloped and retested until every item is resolved. All the changes and retesting are tracked through spread sheets. Applications are not allowed to launch until all identified problem are fixed. Finally, a report is prepared at the end of testing to show exactly what was tested and to list the final outcomes. The software testing methodology is applied in four distinct phases:

- Unit Testing
- Integration Testing
- User Acceptance Testing
- Output Testing

### **6.1.1 UNIT TESTING**

Developers typically do unit testing in order to trace out bugs in each module of the code. Unit testing is done in parallel with coding. It includes testing each function and procedures. Unit testing is also called as module testing. In module testing each module are tested for any possible logical error. They are also tested for specification to see if they are working as per the program should do and they are tested under various conditions. Each module is being tested thoroughly in order to discover pitfalls. Specification testing examines the specification what the program should do and how it should perform under various conditions. The testing will be done by entering data into different tables using forms. The data with less validation will be tested first. Whenever an error is encountered, an informative error message will be displayed which informs user

about the type of error. After the completion of form testing the program will be tested. The unit testing is done to identify

- The image entries are in the correct format.
- No duplicate entries are present.
- To check whether it provide the required result.

### **6.1.2 INTEGRATION TESTING**

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). The task of the integration test is to check that components or software applications, interacts without error. Therefore, testing the data flow between 2 modules is integration testing.

### **6.1.3 USER ACCEPTANCE TESTING**

User acceptance testing is done by the user to check whether the project has met the requirement that has been mentioned at the beginning of the project. Flood alert is tested by the user by inputting values and the result generated is also validated. If the accuracy is as expected user approve the system

### **6.1.4 OUTPUT TESTING**

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. The output generated or displayed by the system under consideration is tested asking the users about the format required by

them. Here, the output is considered in two ways, one is on the screen and other is printed format. The output format on the screen is found to be correct as the format designed according to the user needs. For the hard copy also, the output comes out as specified by the user. Hence output testing does not result in any connection in the system.

## 6.2 VALIDATION CHECK

The validation phase reveals the failures and the bugs in the developed system. It will become known about the practical difficulties the system faces when operated in the true environment. Validation is the process of ensuring that user input is clean, correct, and useful. Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid email?
- Has the user entered text in a numeric field?

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server. Most often, the purpose of validation is to ensure correct user input. Validation can be defined by many different methods, and deployed in many different ways. Server side validation is performed by a web server, after input has been sent to the server. Client side validation is performed by a web browser, before input is sent to a web server.

# **Chapter 7**

## **SYSTEM MAINTENANCE AND FUTURE ENHANCEMENTS**

### **7.1 SYSTEM MAINTENANCE**

System maintenance is a going activity, which covers a wide variety of activities including, removing program and design errors, updating documentation and test data and updating user support system maintenance is a catchall term used to describe various forms of computer or server maintenance required to keep a computer system running properly, it can describe network maintenance which could mean that servers are being physical repaired, replaced or mode. For the purpose of convenience, maintenance may be categorized into three classes they are:

#### **CORRECTIVE MAINTENANCE**

This type of maintenance implies removing errors in a program, which might have kept in the system due to faulty design or wrong assumption.

## **ADAPTIVE MAINTENANCE**

In adaptive maintenance program functions are changed to enable the information system to satisfy the information needs of the user.

## **PERFECTIVE MAINTENANCE**

In perfective maintenance means adding new programs or modifying the existing programs to enhance the performance of the information system. This type of maintenance undertaken to respond to user addition needs which may be due to the changes within or outside of the organization.

## **7.2 FUTURE ENHANCEMENTS**

In future I can expect the modified version of Facebook. The sys-team is very flexible for further up gradation with additional requirement of the self-working, the jsp and hides server makes this Medications very easily. It is also possible to involve more functions into the sys-team. This flexibility makes this system widening its scope. All day-to-day work can be done with much more ease and decency. The database and the information can be updated to the latest coming versions. There are also possibilities for enhancing and further developing the project with the latest information and needs of the portal.

# **Chapter 8**

## **CODING**

### **8.1 SOURCE CODE**

```
package com.syntax.agrobazar;
import android.Manifest;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

```
import android.widget.Toast;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.syntax.agrobazar.CUSTOMER.Customer;
import com.syntax.agrobazar.DELIVERYBOY.DeliveryBoy;
import com.syntax.agrobazar.FARMER.Farmer;
import java.util.HashMap;
import java.util.Map;
public class LoginActivity extends AppCompatActivity {
    EditText username, password;
    Button btnlog;
    TextView reg, login;
    String UNAME, PASS;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        ActionBar actionBar = getSupportActionBar();
        actionBar.hide();

        int PERMISSION_ALL = 1;
```

```
String[] PERMISSIONS = {
    Manifest.permission.WRITE_EXTERNAL_STORAGE,
    Manifest.permission.READ_EXTERNAL_STORAGE,
    Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION,
    Manifest.permission.CAMERA,
    Manifest.permission.READ_SMS,
    Manifest.permission.SEND_SMS,
    Manifest.permission.READ_PHONE_STATE
};

if (!hasPermissions(this, PERMISSIONS)) {
    ActivityCompat.requestPermissions(this, PERMISSIONS,
        PERMISSION_ALL);
}

username = findViewById(R.id.login_username);
password = findViewById(R.id.login_password);
btnlog = findViewById(R.id.login_btnlog);
reg = findViewById(R.id.login_signup);
btnlog.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        UNAME = username.getText().toString();
        PASS = password.getText().toString();
        if (UNAME.isEmpty()) {
            username.requestFocus();
        }
    }
});
```

```
username.setError("enter username");
} else if (PASS.isEmpty()) {
password.requestFocus();
password.setError("enter password");
} else {
volly_login();
// startActivity(new Intent(getApplicationContext(),
UserHome.class));
}
}
});

reg.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
startActivity(new Intent(getApplicationContext(),
RegistrationActivity.class));
}
});
}

@Override
public void onBackPressed() {

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Are you sure you want to exit?")
.setCancelable(false)
.setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
public void onClick(DialogInterface dialog, int id) {
```

```
 LoginActivity.this.finish();
}
})
.setNegativeButton("No", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int id) {
        dialog.cancel();
    }
});
AlertDialog alert = builder.create();
alert.setTitle("AgroBazar");
alert.show();
public void volly_login() {

    RequestQueue queue = Volley.newRequestQueue
        (getApplicationContext());
    StringRequest request = new StringRequest(Request.Method.POST,
        Utility.SERVERUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("*****", response);
            if (!response.trim().equals("failed")) {

                String data = response;
                String arr[] = data.trim().split("\\n");
                SharedPreferences.Editor editor =
                    getSharedPreferences("SharedData", MODE_PRIVATE).edit();
                editor.putString("u_id", "" + arr[0]);
                editor.putString("type", "" + arr[1]);
            }
        }
    });
}
```

```
editor.commit();
//Toast.makeText(Login.this,arr[0],
Toast.LENGTH_SHORT).show();
if(arr[1].equals("CUSTOMER")){
    startActivity(new Intent(getApplicationContext(),
Customer.class));
    Toast.makeText(getApplicationContext(), "Login
Successful", Toast.LENGTH_SHORT).show();

}
} else if (arr[1].trim().equals("FARMER")) {
    startActivity(new Intent(getApplicationContext(),
Farmer.class));
    Toast.makeText(getApplicationContext(), "Login
Successful", Toast.LENGTH_SHORT).show();
} else if (arr[1].trim().equals("DELIVERY_BOY")) {

    startActivity(new Intent(getApplicationContext(),
DeliveryBoy.class));
    Toast.makeText(getApplicationContext(), "Login
Successful", Toast.LENGTH_SHORT).show();
} else {
    Toast.makeText(getApplicationContext(), "Something Went
Wrong",
    Toast.LENGTH_LONG).show();
}
} else {
    Toast.makeText(getApplicationContext(), "Login failed..!",
```

```
        Toast.LENGTH_LONG).show();
    }
}
}, new Response.ErrorListener() {

    @Override

    public void onErrorResponse(VolleyError error) {

        Toast.makeText(getApplicationContext(),
                "my error :" + error,
        Toast.LENGTH_LONG).show();
        Log.i("Myerror", "" + error);
    }
}

@Override

protected Map<String, String> getParams()
throws AuthFailureError {
Map<String, String> map = new HashMap<String, String>();
map.put("key", "login");
map.put("username", UNAME);
map.put("password", PASS);

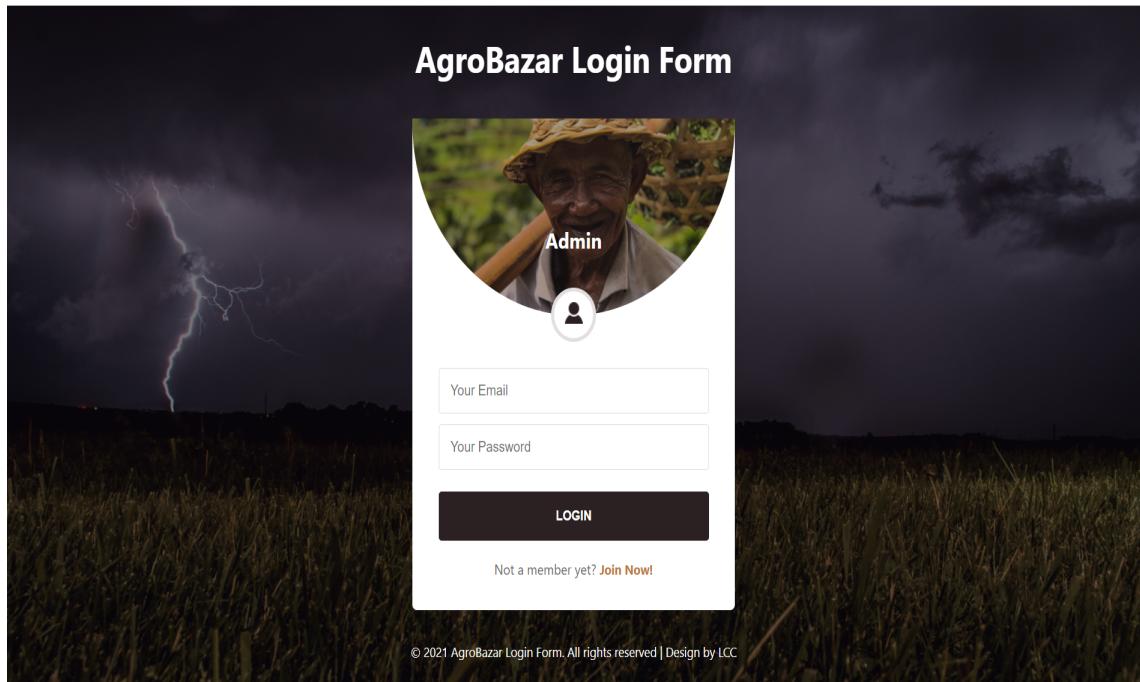
return map;
};

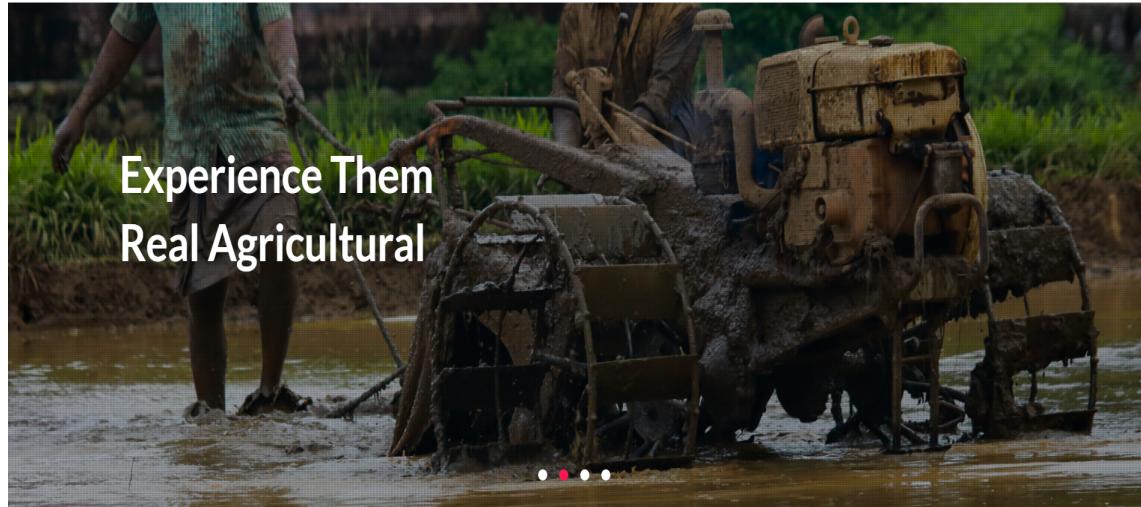
queue.add(request);
}
```

```
public static boolean hasPermissions(Context context, String...  
permissions) {  
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M  
context != null permissions != null) {  
  
for (String permission : permissions) {  
if (ActivityCompat.checkSelfPermission(context, permission) !=  
PackageManager.PERMISSION_GRANTED) {  
return false;  
}  
}  
}  
}  
return true;  
}  
  
}
```

## 8.2 SCREENSHOTS

### 8.2.1 ADMIN PAGE





**Agro Bazar**

Home Farmer Request User Booking Request DeliveryBoy Request Add Category Feedbacks

Logout ⚙

Experience Them  
Real Agricultural

HERE'S WHAT WE DO

---

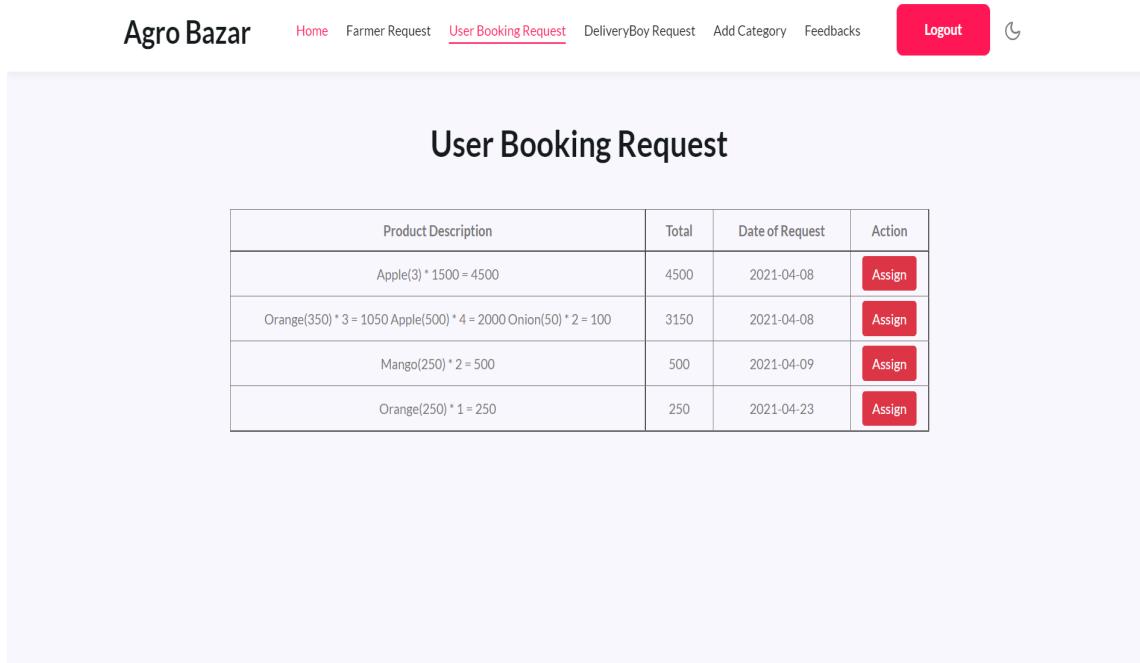
**Agro Bazar**

Home Farmer Request User Booking Request DeliveryBoy Request Add Category Feedbacks

Logout ⚙

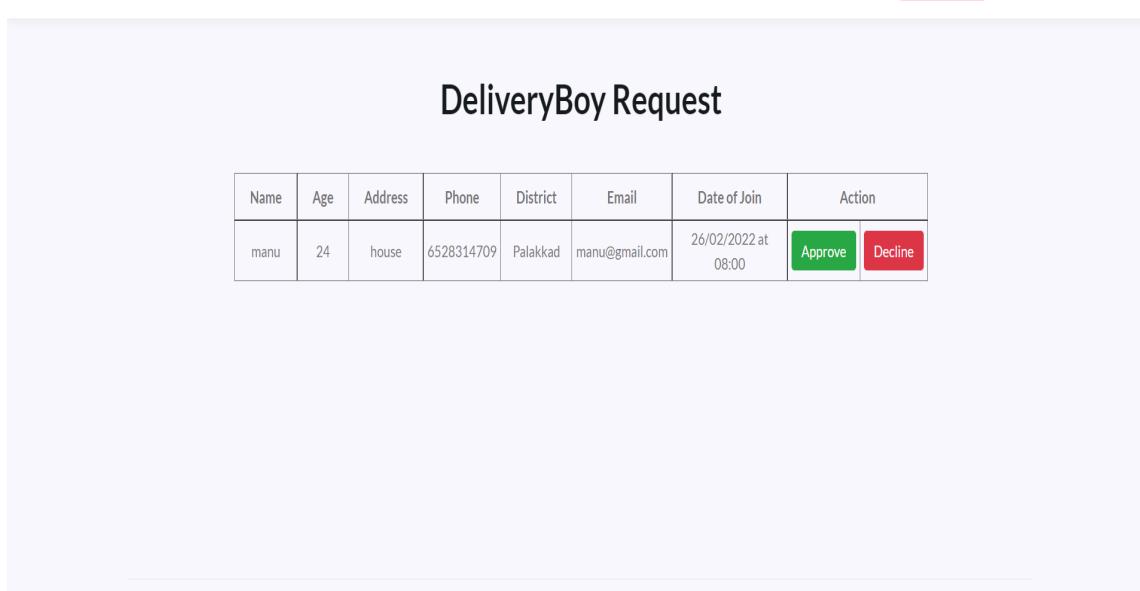
### Farmer Request

Name	Age	Address	Phone	District	Email	Date of Join	Action
Hussain	29	Malappuram	9867866688	Malappuram	hussain@gmail.com	09/04/2021 at 08:25	<button>Approve</button> <button>Decline</button>
shilpa	23	paravur	2583691582	Malappuram	shilpa@gmail.com	01/02/2022 at 10:55	<button>Approve</button> <button>Decline</button>



The screenshot shows the 'User Booking Request' section of the Agro Bazar application. At the top, there is a navigation bar with links: Home, Farmer Request, User Booking Request, DeliveryBoy Request, Add Category, Feedbacks, Logout, and a user icon. The main content area is titled 'User Booking Request' and displays a table of pending booking requests:

Product Description	Total	Date of Request	Action
Apple(3) * 1500 = 4500	4500	2021-04-08	<button>Assign</button>
Orange(350) * 3 = 1050 Apple(500) * 4 = 2000 Onion(50) * 2 = 100	3150	2021-04-08	<button>Assign</button>
Mango(250) * 2 = 500	500	2021-04-09	<button>Assign</button>
Orange(250) * 1 = 250	250	2021-04-23	<button>Assign</button>

The screenshot shows the 'DeliveryBoy Request' section of the Agro Bazar application. At the top, there is a navigation bar with links: Home, Farmer Request, User Booking Request, DeliveryBoy Request, Add Category, Feedbacks, Logout, and a user icon. The main content area is titled 'DeliveryBoy Request' and displays a table of pending delivery requests:

Name	Age	Address	Phone	District	Email	Date of Join	Action
manu	24	house	6528314709	Palakkad	manu@gmail.com	26/02/2022 at 08:00	<button>Approve</button> <button>Decline</button>

The screenshot shows the 'Add Category' page of the Agro Bazar application. At the top, there is a navigation bar with links: Home, Farmer Request, User Booking Request, DeliveryBoy Request, Add Category, and Feedbacks. On the far right of the navigation bar are 'Logout' and a user icon. The main content area has a title 'Add Category'. Below the title is a form with a text input field labeled 'Category Name' and a red 'ADD' button. Below the form is a table listing existing categories:

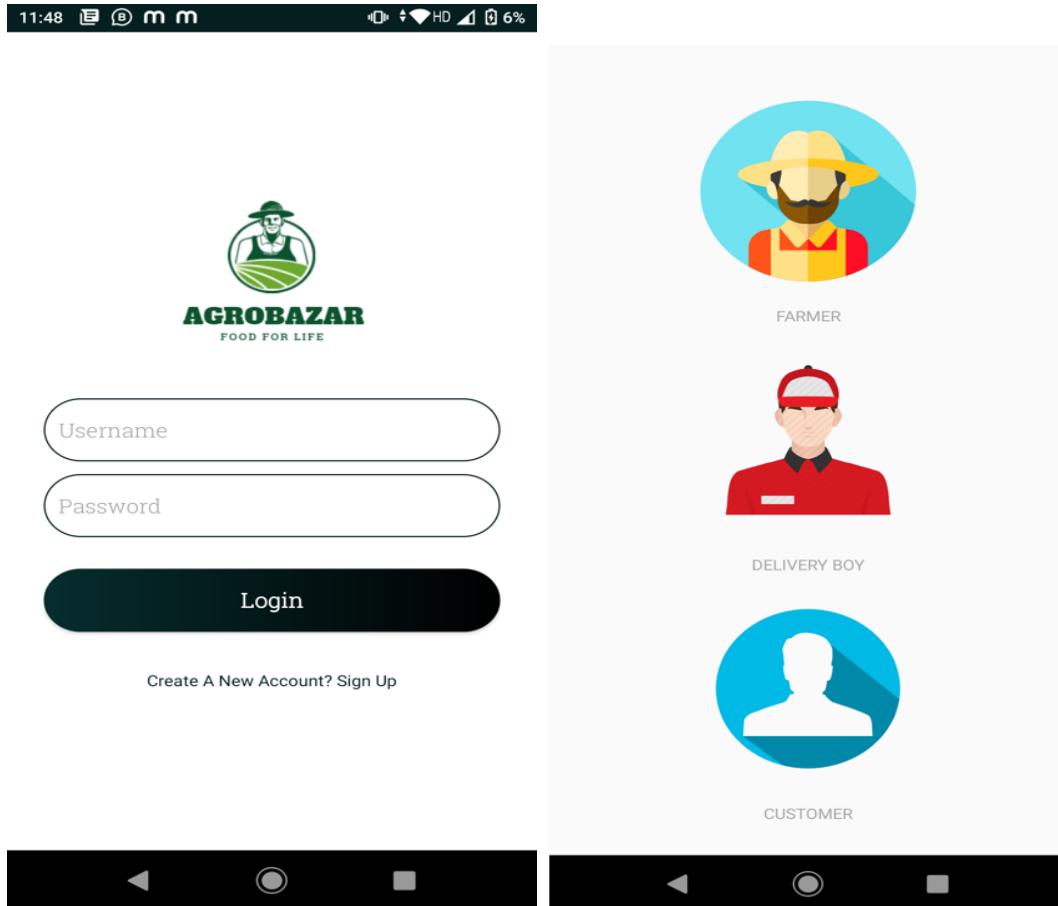
Category Name	Action
Vegetables	Delete
Fruits	Delete
Seeds	Delete

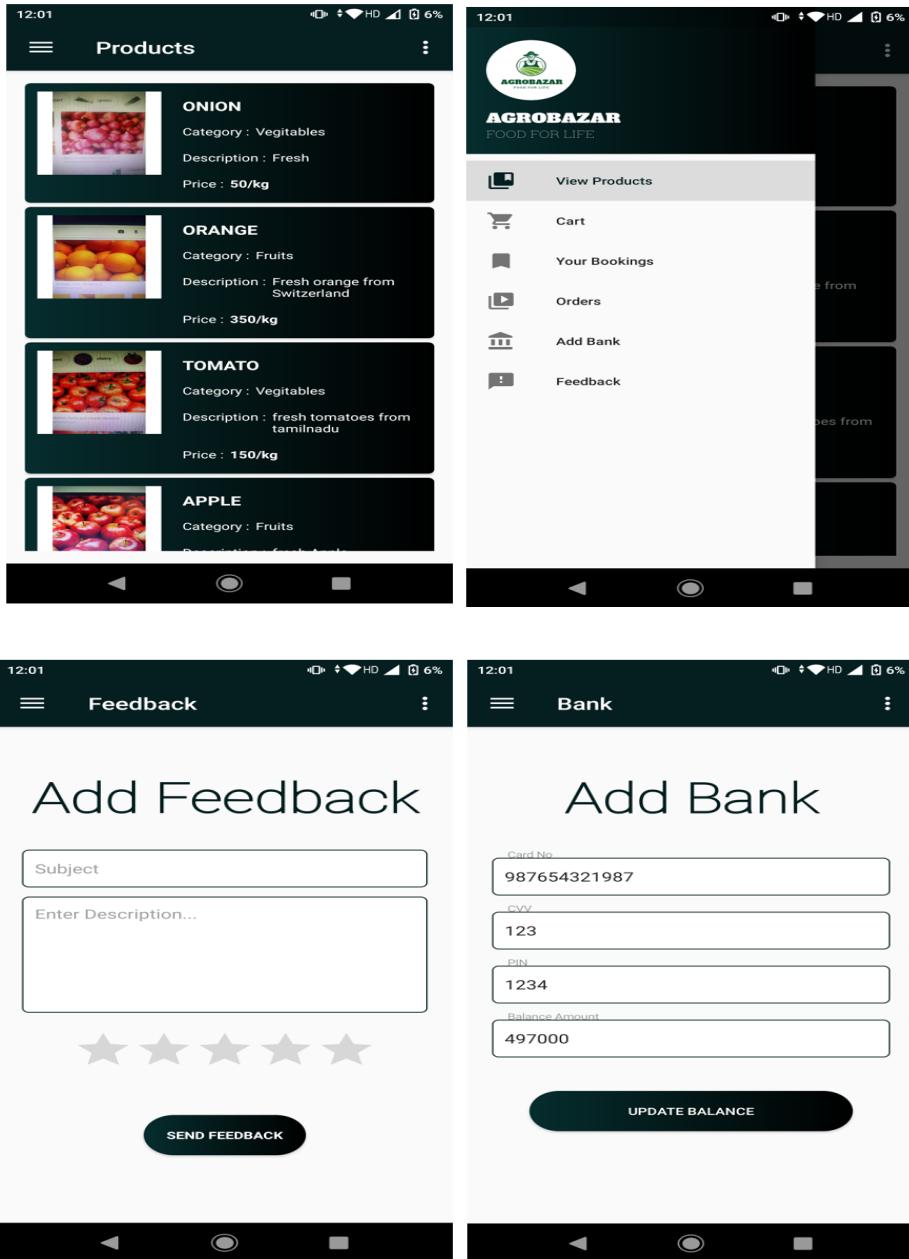
The screenshot shows the 'Feedbacks' page of the Agro Bazar application. At the top, there is a navigation bar with links: Home, Farmer Request, User Booking Request, DeliveryBoy Request, Add Category, and Feedbacks. On the far right of the navigation bar are 'Logout' and a user icon. The main content area has a title 'Feedbacks'. Below the title is a table listing feedbacks:

Name	Phone	Email	Feedback	Rating	Date of Feedback
Akhil	9687876778	akhil@gmail.com	Nice	yfygi	5.0
Amal	9858577759	amalx@gmail.com	Ftxecij	evyvxruh	0
anagha	4526338369	anagga@gmail.com	grap	super	4.0
anagha	4526338369	anagga@gmail.com	tomato	good quality	4.0

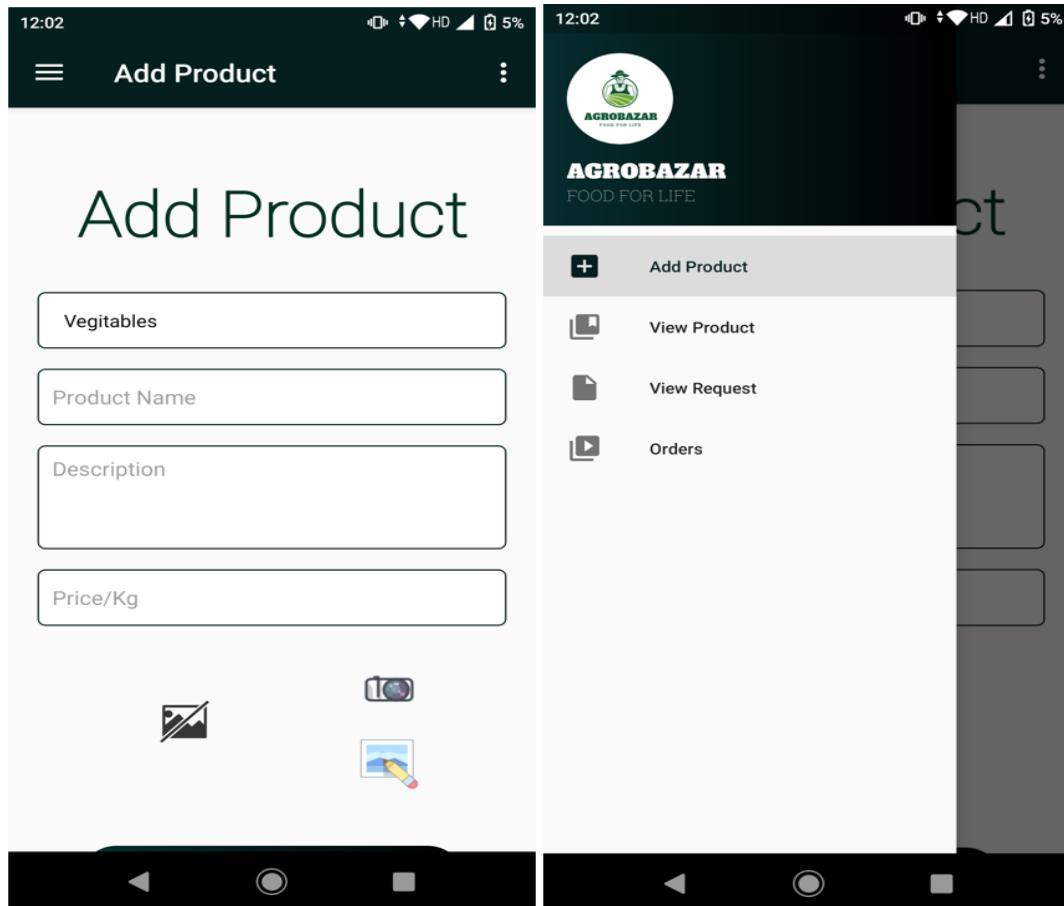
### 8.2.2 LOGIN PAGE



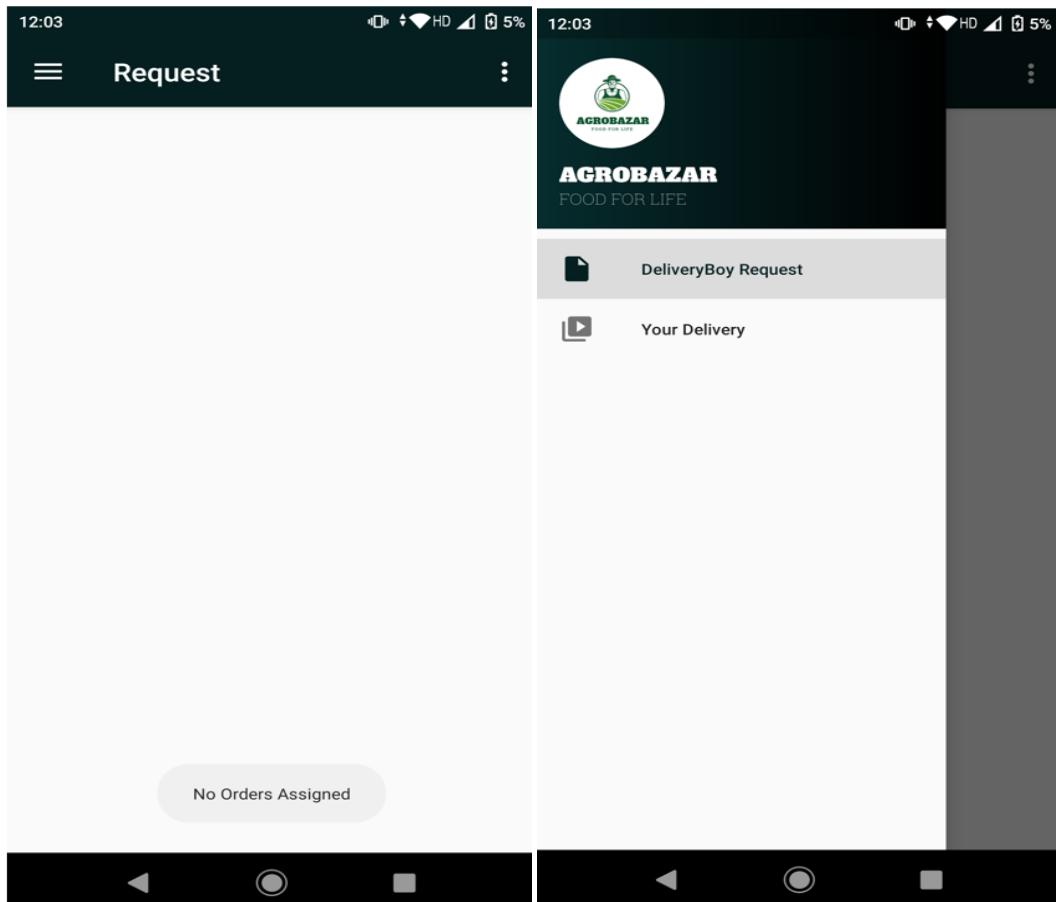
### 8.2.3 CUSTOMER PAGE



### 8.2.4 FARMER PAGE



### 8.2.5 DELIVERY BOY PAGE



# **Chapter 9**

## **CONCLUSION**

The project was successfully completed within the time span allotted. The drawbacks of the existing system as listed before are fully evacuated. All the existing inconsistencies are fully solved as this system is implemented. This reduced the burden of the administration of the system. All the modules are tested separately and put together to form the main system. Finally, the system is tested with real data and it worked successfully. Thus, the system has fulfilled the entire objective defined. The system has been developed in an interactive manner; the reports generated by the system are clear. The system is flexible, user friendly and has its own full data security and all data recovery facility. The developed system has mainly two modules admin and the common Facebook user.

### **9.0.1 BIBLIOGRAPHY**

- <http://developer.android.com/guide/index.html>.
- <http://developer.android.com/reference/packages.html>
- <http://www.youtube.com>
- <http://developer.android.com/guide/components/fundamentals.html>
- <http://developer.android.com/guide/topics/ui/index.html>
- <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- <http://developer.android.com/guide/appendix/faq/commontasks.html>
- <https://ssrn.com/abstract=3847520>
- <http://dx.doi.org/10.2139/ssrn.3847520>
- <http://code.google.com/android/add-ons/google-apis/maps-overview.html>
- <http://developer.android.com/resources/samples/get.html>