

Dairy Dreaming And Complain Service Facility App

By

Nauman Shehzad

Awais Hafeez



Department of Computer Sciences

Comsats University Islamabad

Attock Campus (Pakistan)

Session 2020-2021

Dairy Dreaming And Complain Service Facility App



Undertaken By:

Nauman Shehzad

CIIT/FA19-MCS-007

Awais Hafeez

CIIT/FA19-MCS-016

Supervised by:

Miss Mehreen Wahab

A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE
ENGINEERING

**DEPARTMENT OF COMPUTER SCIENCES
COMSATS UNIVERSITY ISLAMABAD,
ATTOCK CAMPUS – PAKISTAN**

SESSION 2020-2021

Declaration

We hereby declared that this application, neither whole nor as a part has been copied out from any source. This whole application is plagiarism-free. It is further declared that we have developed this project and accompanied report/documentation entirely based on our efforts, talents, techniques, and skills without taking help from any illegal way. If any part of this application is proved to be copied out from any source, considered to be plagiarism in the report, or found to be a reproduction of some other source, we are fully responsible for the consequences of this.

Nauman Shehzad

(FA19-MCS-007)

Awais Hafeez

(FA19-MCS-016)

Dated:

FINAL APPROVAL

It is certified that we have read this project (Dairy Dreaming and Complain Service Facility App) report submitted by **Mr. Nauman Shehzad (FA19-MCS-007)** and **Mr. Awais Hafeez (FA19-MCS-016)**. It is, in our judgment, of sufficient standard to warrant its acceptance by the Department of Computer Science, Comsats University Islamabad, Attock Campus.

Committee:

1. External Examiner _____
(Examiner Name)
Designation
University Name
2. Supervisor _____
Miss Mehreen Wahab
3. Chairperson _____
(Chairperson Name)
4. Head of Department _____
(HoD Name)

ACKNOWLEDGEMENT

“First of all, starts with the name of **Allah**, the most merciful and kind.”

We are thankful to **Allah Almighty** who provides uses all the resources, techniques, and others to help us in developing an application that is useful for humanity. May **Allah** helps us in developing all such things which are beneficial for humanity. After that, we are thankful to our Supervisor **Miss Mehreen Wahab** who helped us continuously throughout our project. She helped us in solving every difficulty while we faced in our project. We always found her very helpful and kind throughout our project. She also helps us in discussing different ideas and features about our project. We are also thankful to our parents, friends, relatives, and all others that help us in solving every problem we faced in our project, both financially and morally. I would also like to thank the technical helpers and professionals for their great support and help. Without having the support of all that mentioned above, we are nothing and would not develop this project.

Thanks to all.

PROJECT BRIEF

PROJECT NAME	DAIRY DREAMING AND COMPLAIN SERVICE FACILITY APP
ORGANIZATION NAME	COMSATS UNIVERSITY ISLAMABAD ATTOCK CAMPUS
OBJECTIVE	PARTIAL FULFILMENT FOR THE DEGREE OF MASTERS IN COMPUTER SCIENCE.
UNDERTAKEN BY	NAUMAN SHEHZAD & AWAIS HAFEEZ
SUPERVISED BY	MISS MEHREEN WAHAB LECTURER COMPUTER SCIENCE COMSATS UNIVERSITY ISLAMABAD ATTOCK CAMPUS
STARTED ON	15-SEPT-2020
COMPLETED ON	CONTINUE
COMPUTER USED	DELL LAPTOP (8GM RAM, 320GB HARD DISC)
SOURCE LANGUAGE	JAVA
OPERATING SYSTEM	WINDOWS
TOOLS USED	ANDROID STUDIO, STARUML, FIGMA.

ABSTRACT

The application “Dairy Dreaming and Complain Service Facility App” automates the whole process of selling and buying Dairy products i.e., Milk and Animals from the farm. It aims to facilitate the farm, the user (buyer), and all the actors participating in the system. The “Dairy Dreaming and Complain Service Facility App” is a completely android based application that helps the user, the farm owner, and other all actors to communicate with each other for the buying and selling of the products. All the actors participating in the system will have their proper account to the system through which they enter the system via sign-in/sign up and upload info according to their degree of nature.

The Farm and the User can find each other’s location via “Google Map” which provides the locations for communication. This application is a completely Android-based Mobile application and is available free of cost to every user who wishes to use it. The basic aims for developing this application are to find a better and an easy way the communicate between the different actors of the system like Farm, User, etc.

Table of Contents

Chapter 1:	1
Introduction	1
1.1. Introduction	2
1.2. Objectives	3
1.2.1. Cost-Effective and Time-saving.....	3
1.2.2. Responsive	3
1.3. Problem Statement	3
1.4. Proposed Solution	3
1.5. Scope of the System	4
1.6. Hardware Requirements	4
1.7. Software Requirements	4
1.8. Tools	4
1.8.1. Android Studio	4
1.8.2. Figma	4
1.8.3. Firebase	5
1.8.4. StarUML	5
Chapter 2:	6
Literature Review	6
2.1. Literature Review	7
2.2. Existing System	7
2.3. Proposed System	7
2.3.1. Comparison between Existing System and Proposed Solution	8
Chapter 3:	9
Requirement Specification	9
3.1. Requirement	10
3.2. Functional Requirements	10
3.2.1. Sign Up	10

3.2.2. Sign In	11
3.2.3. User Account	11
3.2.4. Farm Owner Account	12
3.2.5. Update Data	12
3.2.6. Delete Data	13
3.3. Non-Functional Requirements	13
3.3.1. Performance	13
3.3.2. Reliability	14
3.3.3. Accuracy	14
3.3.4. Availability	14
3.3.5. User Friendly	15
3.3.6. Security	15
3.3.7. Portability	15
Chapter 4:	16
Design System	16
4.1. Design System	17
4.2. Use Case Diagram	17
4.2.1. User Use case Diagram	18
4.2.2. Farm Owner Use case Diagram	19
4.2.3. Food Authority Use case Diagram	20
4.3. Activity Diagram	21
4.3.1. Farm Owner Activity Diagram	21
4.3.2. User Activity Diagram	22
4.4. Data Flow Diagram	23
4.4.1. 0-Level DFD of the System	23
4.4.2. 1-Level DFD of the System	24
4.5. Class Diagram	25
4.5.1. Class Diagram of the System	25
4.6. Sequence Diagram	26
4.6.1. Farm Owner Sequence Diagram	26
4.6.2. User Sequence Diagram	27

Chapter 5: 28

Implementation 28

5.1. Implementation	29
5.2. Tools and Technologies	29
5.2.1. Tools	29
5.2.2. Language	29
5.2.3. Data Access	29
5.3. Software Requirements	29
5.4. Hardware Requirements	29
5.5. User Interface	29
5.5.1. Splash Screen	30
5.5.2. Switch Board	30
5.5.3. Sign Up Activities	31
5.5.4. Sign In Activities	31
5.5.4.1. Sign In (1)	31
5.5.4.2. Sign In (2)	32
5.5.5. Associated Action for the User	32
5.5.6. Lists of all available Farms	33
5.5.7. Ordering of Milk via Offline or Online delivery	33
5.5.8. Ordering of Animal via Offline or Online delivery.....	34
5.5.9. Location (dependent on Delivery method)	34
5.5.10. Location of the Farm (in case of Online delivery)	35
5.5.11. Location of the User (in case of offline delivery)	35
5.5.12. Associated Action for Farm Owner	36
5.5.13. Upload Data	36
5.5.14. Update or Upload Milk	37
5.5.4.1. Update or Upload Price of each Milk type	37
5.5.4.2. Update or Upload Stock of each Milk type.....	38
5.5.15. Upload Animal of each type	38
5.5.16. Upload Animal Info	39
5.5.17. Manage orders of Milk of the Users.....	39
5.5.18. Manage orders of Animals of the Users	40

5.5.19. Lists of Farms that are complained by the User.....	40
5.5.20. Complain Form for the User	41
Chapter 6:	42
Evaluation	42
6.1. Evaluation	43
6.2. Testing	43
6.2.1. Unit Testing	43
6.2.2. Integration Testing	43
6.2.3. System Testing	43
6.3. Test Cases	43
6.3.1. Test Case 1 (Sign Up)	43
6.3.2. Test Case 2 (Sign In)	44
6.3.3. Test Case 3 (Ordering of Milk)	44
6.3.4. Test Case 4 (Uploading of Milk)	44
6.3.5. Test Case 5 (Uploading of Animals)	45
Chapter 7:	46
Conclusion and Future Work	46
7.1. Conclusion and Future Work	47
7.1.1. Conclusion	47
7.1.2. Future Work	47
Chapter 8:	48
References	48
8.1. References	49

List of Figures

Figure No.	Figure Title	Page No.
4.1	User Use case Diagram	18
4.2	Farm Use case Diagram	19
4.3	Food Authority Use case Diagram	20
4.4	Farm Owner Activity Diagram	21
4.5	User Activity Diagram	22
4.6	Data Flow Diagram (Level-0) of the System	23
4.7	Data Flow Diagram (Level-1) of the System	24
4.8	Class Diagram of the System	25
4.9	Farm Owner Sequence Diagram	26
4.10	User Sequence Diagram	27
5.1	Splash Screen	30
5.2	Switch Board	30
5.3	Sign Up Activities	31
5.4	Sign In Activities (1)	31
5.5	Sign In Activity (2)	32
5.6	Associated Action for the User	32
5.7	List of all Available Farms	33
5.8	Ordering of Milk	33
5.9	Ordering of Animals	34
5.10	Location Option	34
5.11	Location of the Farm	35
5.12	Location of the User	35
5.13	Associated Action for Farm Owner	36
5.14	Upload Data	36
5.15	Update Price and Stock	37
5.16	Update, Upload Price of Milk	37
5.17	Update, Upload Stock of each Milk type	38
5.18	Upload Animal	38
5.19	Upload Animals' Info	39
5.20	Manage Orders of Milk	39
5.21	Manage Orders of Animals	40
5.22	Lists of all Complained Farms	40
5.23	Complain form for the User	41
5.24	Milk Data of Selected Farm	41

List of Tables

Table No.	Table Title	Page No.
2.1	Comparison between Existing System and Proposed System	8
3.1	FR Sign Up	10
3.2	FR Sign In	11
3.3	FR User's Account	11
3.4	FR Farm Owner Account	12
3.5	FR Update Data	12
3.6	FR Delete Data	13
3.7	NF Performance	13
3.8	NF Reliability	14
3.9	NF Accuracy	14
3.10	NF Availability	14
3.11	NF User Friendly	15
3.12	NF Security	15
3.13	NF Portability	15
4.1	User Use case Table	18
4.2	Farm Owner Use case Table	19
4.3	Food Authority Use case Table	20
6.1	Test Case 1 (Sign Up)	43
6.2	Test Case 2 (Sign In)	44
6.3	Test Case 3 (Ordering of Milk)	44
6.4	Test Case 4 (Uploading of Milk)	44
6.5	Test Case 5 (Uploading of Animal)	45

Chapter 1

Introduction

1.1. Introduction:

Milk is an essential ingredient of life. It is not available at such amount which is sufficient for everyone. Most people do not have a transport facility to visit the farm or if someone has a transport facility then there is a possibility that he/she returns empty hands due to enough milk on the farm. Most people also wish to buy animals for their own needs from the farm so the info of animals must also be provided to the user. The user must have an option to buy offline delivery on online home delivery with extra charges. The quality and standard of milk must be ensured by the farm owner. Otherwise, the user may complain to Food Authority for the bad standard of product. All these problems and difficulties are going to be solved in the app that we are developing. This application will provide the facility to buy milk and animal of a particular kind via offline or online delivery. The user either a simple customer or wholesaler will enter his/her request through his console via sing in or sign-up. The user must enter his/her Name, Email, Password, and Confirm Password for registration, and for sign in, the user must enter only his/her Email and Password. We also have different types of farm owners with different provided locations. Thus, the farm owner also must sign in or sign up to the app for providing the details to the users. The location of the farm and the user both will be provided so that for ease in offline or online via Google map. The service will be available 24/7 hours a day. In the case of online delivery, some extra charges (Rs.100 for Milk per type and Rs.500 per Animal type) will be taken from the user and for the offline delivery no extra charges will require and a Token No. will be generated for the user. The most important is the involvement of the Food Authority. If the user is not satisfied with the quality of milk, then he/she can complain to Food Authority for the bad standard and quality. The Food Authority will have a list of all the farms that are complained about by the user. The food authority will sign in to the app to act. Four types of Food Authorities are included in the application as province type that is Punjab, Balochistan, KPK, and Sindh. Then the Food Authority will take serious action against a particular farm either imposing a huge fine or completely seal that farm. The user simultaneously can get milk and animal from the particular farm. The application has four modules namely: Simple Customer, Wholesaler, Farm Owner, and Food Authority. The Wholesaler will get some advantage as compared to Simple Customer i.e., Rs.10 less per kg for Milk and Rs.1000 less per Animal. The last order info is also included in the application.

1.2. Objective:

The objective of this app is based on Cost-effective and time saving and Responsiveness.

1.2.1. Cost-Effective and Time saving:

The (Our) application is saving time for the user as well as the farm owner in an efficient manner. If the user's request is not fully fulfilled from a particular farm, then he/she can quickly get the required response from the farm. The user can save his/her time via online delivery and cost via offline delivery. Both participating parties (the user and the farm owner) need communication between them which will be done through this app. The involvement of the Food Authority is increasing the quality and the standard of the Milk. All these facilities of the app will be cost-effective and time-saving for the participating parties.

1.2.2. Responsiveness:

We are using the DBMS named Firebase for maintaining the record of all the modules using the real-time database. Firebase is much faster and efficient than other DBMS for handling records. It is very crucial for the system for maintaining all the records and make communication between the users and the farm owners. The system will respond as soon as possible for ease of communication between the user and the farm owner.

1.3. Problem Statement:

There are many problems for the people related to the Dairy System. The users cannot find a proper farm that fulfills the user's request and vice versa. Some people cannot have any transport facility for buying Milk or Animal. The user cannot find the acknowledgment of his/her request from home. Time is the most important factor in this area which must need to be saved. The user should have an option for offline or online delivery for receiving an order of his own choice. Both the farm owner and the user need a way of communication between them. The locations of both the user and the farm are required for this.

1.4. Proposed Solution:

Communication will be provided between the users and the farm owners via location. The farms are providing the facility of both online and offline delivery to their users. The location of the farm is required for online delivery and the location of the user will be required for

offline delivery. Hence, the locations of both the users and the farms are required in this application. For ensuring the quality of the Milk, users have an easy solution to complain to Food Authority.

1.5. Scope of the System:

Whenever the user gives his/her request or order to the Farm, the availability/unavailability of his/her request will be checked by the owner of the Farm. In case of availability, the acknowledgment to the user will be sent otherwise the user cannot get his/her order. All the data will be stored in the firebase i.e., the real-time database which required a stable Internet connection.

1.6. Hardware Requirements:

The application “Dairy Dreaming and Complain Service Facility App” is completely android base. The communication between the parties is done via Google Maps. The application required an Android Smartphone or Emulator with a stable Internet connection for work. For testing or running any app developed in Android Studio, you required an Android Emulator or Android Smartphone.

1.7. Software requirements:

Android version of the device must be a lollipop or above this.

1.8. Tools:

1.8.1. Android Studio:

Android Studio is the official Integrated Development Environment (IDE) for developing all Android applications. It is a much faster tool for developing Android applications than others and has many built-in features support. In this tool, Java or JavaScript is used for implementing logic, and XML is used for designing the interfaces. It has different kinds of Emulators for testing the app you have developed with different sizes and nature.

1.8.2. Figma:

We use Figma as a backend tool for designing interfaces. All the backgrounds are designed in Figma. The Figma provides many built-in features for designing and creating the background.

You can insert text as well as images. It provides the facilities to draw circles, lines, ellipses, rectangles, etc. of different sizes.

1.8.3. Firebase:

The firebase is used to store all the data of the app as a real-time database. Data is stored as JSON in the firebase. The Firebase allows you to develop a responsive, efficient, accurate app. The firebase requires your E-mail address and Internet connection to be used as a database.

1.8.4. StarUML:

Star UML is a drawing tool that we use to create diagrams for the system being developed. Use cases, Data Flow Diagram, ERD and Activity Diagrams, etc. can be drawn efficiently in an easy way in this tool. Star UML provides many facilities and built-in symbols that make an easy in creating all types of diagrams.

Chapter 2

Literature Review

2.1. Literature Review:

The application “Dairy Dreaming and Complain Service Facility App” automate the whole process of buying of Milk and Animal. It aims to facilitate the users and the farm owners. The users who need Milk or Animal of a particular kind will search for a particular nearest farm based on the location provided by google maps. The Farm owner required the location of the user in case of offline delivery and the user require the location of the farm in case of online delivery. Thus, the locations of both the farm and the user are required in this app. In case of unavailability of the request, the user searches another nearest farm with the help of a Google map. In case of offline, the location of the user is required, and a Token No. will be generated to the user. This application aims to make communication easy and efficient between the users and the farm owners. The application is completely Android-based and will only run on the Android Smartphone or Android Emulator.

2.2. Existing System:

The existing system exists namely “Dairy Product Service” has many deficiencies as it does not provide the selling of Animal of a particular kind while our application is selling Milk as well as Animal. The user cannot interact with the Food Authority to complain in the existing system while in our app the user has the facility to complain to the Food Authority in case of bad standard or bad quality of milk. The other existing system namely “Farm Ghar” is only about selling different kinds of Animals not about selling Milk and is completely online. The existing application is providing only online delivery of Milk while our app is providing online as well as offline delivery with generating of Token No. of Milk and Animal.

2.3. Proposed System:

To overcome the problems of the existing system, we created an application named “Dairy Dreaming and Complain Service Facility App”. Our proposed system has many facilities for both the User (Simple customer or Wholesaler) and the Farm owner. All the modules or actors of the system can register themselves via the same application with different interfaces. Our application is providing the facility for both the User and the Farm owner to easily find each other for communication. The locations of both the user and the farm will be provided with the help of Google Maps for this purpose. The farm owner can easily manage the accounts and

orders of the users via a database. The user can complain to the Food Authority of the particular farm in case of bad standards or quality of Milk. Food Authority may then impose a huge fine on the owner of that farm or completely seal that farm. The user can easily move to another farm if his/her request is not fulfilled in the current farm. The farm owner can add, delete, or update the record of the users in the database. The Food Authority will have a list of all farms that are complained by the User.

2.3.1. Comparison between Existing System (Dairy Product Service) and Proposed System (Dairy Dreaming and Complain Service Facility App):

Features	Dairy Product System	Farm Ghar	Dairy Dreaming and Complaint Service Facility App (Proposed)
Online Delivery	✓	✓	✓
Offline Delivery	✗	✗	✓
Selling Milk of different nature	✓	✗	✓
Selling Animals of different nature	✗	✓	✓
Facility to complain to Food Authority by the user	✗	✗	✓
Generating Token No. automatically for Offline Delivery	✗	✗	✓

Table: 2.1 – Comparison between Existing System(s) and Proposed System

Chapter 3

Requirements Specification

3.1. Requirement:

A Requirement is a need for the proposed system to be built. A Requirement is a statement that one actor demands on another explicitly and implicitly. It is the most essential part and must be met in all relation to full fill the requirements of all the entities involved in the system. Gathering the requirement of a system is the most important part as no other part is as difficult as this one. Simply, it is the statement of a system service or constraint. Requirement engineering is the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.

3.2. Functional requirements:

The statements or services that the system should do are the functional requirements of the system. Functional requirements may also describe what the system should not do. Functional requirements are statements that must be clear, unambiguous, and consistent. They are written in such a language that should be easy to read and understand and is highlighting their purpose without creating any confusion.

3.2.1. Sign Up:

Name:	FR-1: Sign Up
Description:	To use this app (first time), the user must have to Sing up to the app to register himself.
Rationale:	Without signing up, no one can use this application. Registering or Signing up is the fundamental part of every system.
Fit Criterion:	<p>For Signing Up, the following information is mandatory for the system.</p> <ul style="list-style-type: none"> • All the fields like Name, Email, Password, and Confirm Password must be empty before the user writes anything in the fields. • Then the user will enter his/her correct Name, Email, Password, and Confirm Password. After authentication, the system will then grant user access. • If the user enters an incorrect Name or Password or both then the system will not grant access to the app to the user.
Dependencies:	If the user does not register or sign up himself to the system, then the other functions will not be performed for the user.

Table: 3.1 – FR Sign Up

3.2.2. Sign In:

Name:	FR-2: Sign In
Description:	If the user wants to use this app many times, he/she has to Sign In to the app by entering his/her correct Email and Password.
Rationale:	The system will perform the authentication technique to ensure whether the entered Email and Password are correct. After authentication and ensuring entered Email and Password are correct, the system will grant access to pre-existing users to the system. If the authentication is incorrect then the system will not grant access to the app.
Fit Criterion:	<p>For Signing In, the user has to enter the following details into the system.</p> <p>i. Email:</p> <p>The data type of Email is String. The user has to enter his/her Correct Email for Signing into the system.</p> <p>ii. Password:</p> <p>The data type of Password is also String. The user can enter any password consisting of alphabets, numbers, or any special character.</p>
Dependencies:	If the user does not perform sign-in to the system, then the other functions will not be performed and the system will not perform any access to the system.

Table: 3.2 – FR Sign In

3.2.3. User's Account:

Name:	FR-3: Managing User's Account
Description:	The farm owner can see the orders of the users and other details. He/She can easily manage their accounts in his database.
Rationale:	The farm owner can see all the orders of the users. For this functionality, the farm owner can see and set the list of all the users who requests their order.
Fit Criterion:	The farm owner can see all the details of the users. He can also update this data in the database and can grant acknowledgment, Total bill, and generated Token no. If the farm owner does not manage these details the management of records will become a serious issue.
Dependencies:	The functionality depends on the users.

Table: 3.3 – FR User's Account

3.2.4. Farm Owner Account:

Name:	FR-4: Farm Owner Account
Description:	The farm owner will upload the data about Milk and the Animals to the user. The farm owner can check the details of all the users i.e., their Name, Password, Order, Bill, Token No. etc. The farm owner also has a record, whether the delivery is offline or online.
Rationale:	When the farm owner uploads all the details about Milk and the Animals then the data will be shown to the user. Then the user can see all info about Milk and the Animals uploaded by the farm owner.
Fit Criterion:	The communication between the farm owner and the user will be done via locations provided with the help of Google Maps. For an offline delivery, the user required the location of the farm and for the online delivery, the farm required the location of the user.
Dependencies:	This functionality depends on the Farm owner.

Table: 3.4 – FR Farm Owner Account

3.2.5. Update Date:

Name:	FR-5: Update Data
Description:	The farm owner can update the record of the users (simple customers and wholesalers) in his database.
Rationale:	Updating the record is necessary because the user can enter wrong info like incorrect Name or Password or both. For accuracy purposes, all the records must be managed efficiently.
Fit Criterion:	For updating the profile information of the user, the user may put his/her information that is a little bit wrong such as the wrong Name or Password and needs to be updated for accuracy purposes.
Dependencies:	This functionality depends on all the actors involved in the application.

Table: 3.5 – FR Upload Data

3.2.6. Delete Date:

Name:	FR-6: Delete Data
Description:	The farm owner can delete the accounts of users (simple customers or Wholesalers) for some purpose or in case of some scenario.
Rationale:	If the user does not respond to the farm after his/her request, then the record of that user needs to be deleted.
Fit Criterion:	If the record of such users who are no longer participating with the system are stored, then this will create an extra burden on the database (firebase).
Dependencies:	Depends on the order or info of the user.

Table: 3.6 – FR Delete Data

3.3. Non-Functional Requirements:

Non-Functional requirements define the attributes of the system such as performance, security, reliability, accuracy, portability, etc.

3.3.1 Performance:

Name:	NF-1: Performance
Description:	Storing and retrieving data must be very fast. The performance is measured in response time.
Rationale:	If the storing and retrieving of data is not fast, then the time wastes and the user will avoid using the application. Moreover, decreasing in performance will decrease the efficiency of the app as well.
Requirement:	If the Internet is available, then the retrieving information would be measured in almost 2 or 3 seconds.

Table: 3.7 – NF Performance

3.3.2 Reliability:

Name:	NF-2: Reliability
Description:	The system should be reliable in the sense that it should not damage or crash due to a load of data in its database.
Rationale:	If the system or app is not reliable then the robustness of the system will be increased and the whole record will be put at risk. The system should not crash in any load of data.
Requirement:	The system should have the best hardware and use the modern best tools and technologies.

Table: 3.8 – NF Reliability

3.3.3 Accuracy:

Name:	NF-3: Accuracy
Description:	All the data stored in the database must be accurate i.e., all the info must be correct and unambiguous.
Rationale:	If the stored information is not accurate then the system will have wrong and inaccurate records of users. Consequently, the app will lose its reputation. The farm owner will not be able to manage all the records.
Requirement:	The system should have specialized tools for storing accurate information.

Table: 3.9 – NF Accuracy

3.3.4 Availability:

Name:	NF-4: Availability
Description:	Make sure this app should be available at any time. The app will be available 24/7 hours a day with a need for an Internet connection
Rationale:	If the system is not available 24/7 hours a day, then it cannot fulfill the need of every user. Therefore, many users will refuse or avoid the use of the app.
Requirement:	The system should be online because of having an online database.

Table: 3.10 – NF Availability

3.3.5 User Friendly:

Name:	NF-5: User Friendly
Description:	Make user this app should be user friendly i.e., it should be easy to use for every person related to the application and do not face any difficulty while using the app.
Rationale:	If the system is not user-friendly, then no one can use this. It will become difficult for the users for using it and consequently, they will avoid using the app.
Requirement:	The system should support only the English language and concise and meaningful words. It should use such words and sentences that are easy to understand for every user related to the system.

Table: 3.11 – NF User Friendly

3.3.6 Security:

Name:	NF-6: Security
Description:	Make user this app should be secure i.e.; only valid and authorized users can use the app via authentication.
Rationale:	If the system is not secure, then its security will be put at risk. Consequently, unauthorized and non-registered users can use the app.
Requirement:	The user must mention his/her correct Name and Password before using the app.

Table: 3.12 – NF Security

3.3.7 Portability:

Name:	NF-7: Portability
Description:	Make user this app should be portable to every user's device. It should run on all Android-based devices.
Rationale:	If the system is not portable, then some users having Android devices with different nature will refuse or avoid using the app.
Requirement:	The system should be designed in such a way that it should run on all Android-based devices.

Table: 3.13 – NF Portability

Chapter 4

Design System

4.1. Design System:

The design system is a software engineering process used to analyze the system being developed. It is concerned with IT where computer system requires complete analysis according to their design and markup. In the analysis phase, all the modules, interface data, etc. are analyzed to satisfy the requirements. This chapter will describe the system design. It will analyze the system using different techniques and diagrams as Activity diagrams, Use cases, Class diagram, Sequence diagram of the system being developed. The best approach is to gather and define the requirement without any ambiguity, conflict, or inconsistency. It removes the uncertainties, risks of the system so that the users can use the app with satisfaction when the final version of the product is delivered to the user.

4.2. Use Case Diagram:

Use a case diagram is used to show the behavior of the system. It is used to identify the system being developed graphically. Use cases use Actors to identify the system. Actors may be external to the system or internal to the system. It is used to identify the boundary of the complete system being developed. They are four different actors used in our system:

- Farm Owner.
- Simple Customer.
- Wholesaler.
- Food Authority.

4.2.1. User Use case Diagram:

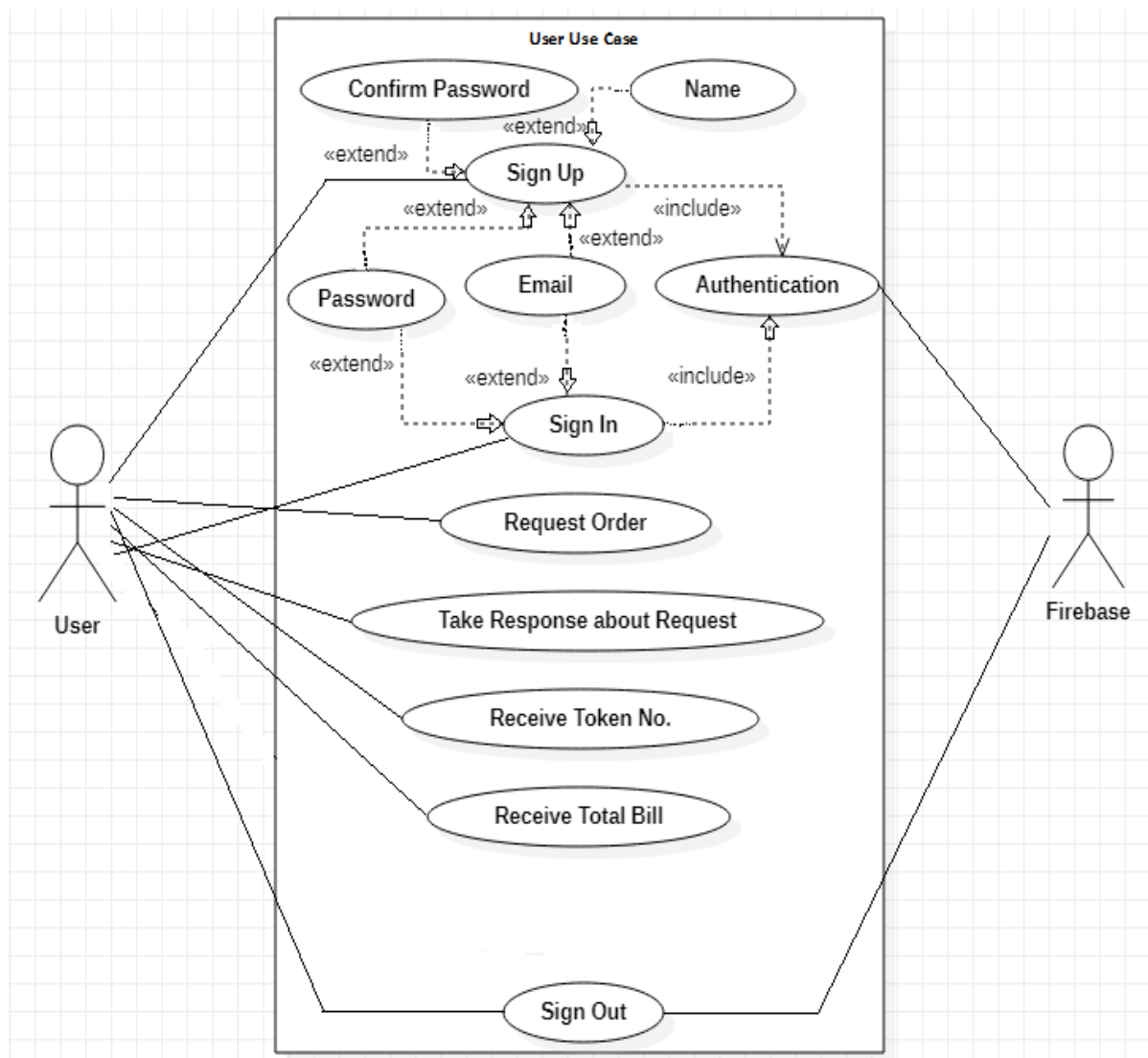


Figure: 4.1 – User Use case Diagram

Use Case Id	UC-1
Actor	User
Pre-Condition	Sign in/Sign up to the system
Functionality	<ul style="list-style-type: none"> • Request Order • Receive Token No. • Receive Total Bill.
Post-Condition	Sign out from the system

Table: 4.1 – User Use case Table

4.2.2. Farm Owner Use case Diagram:

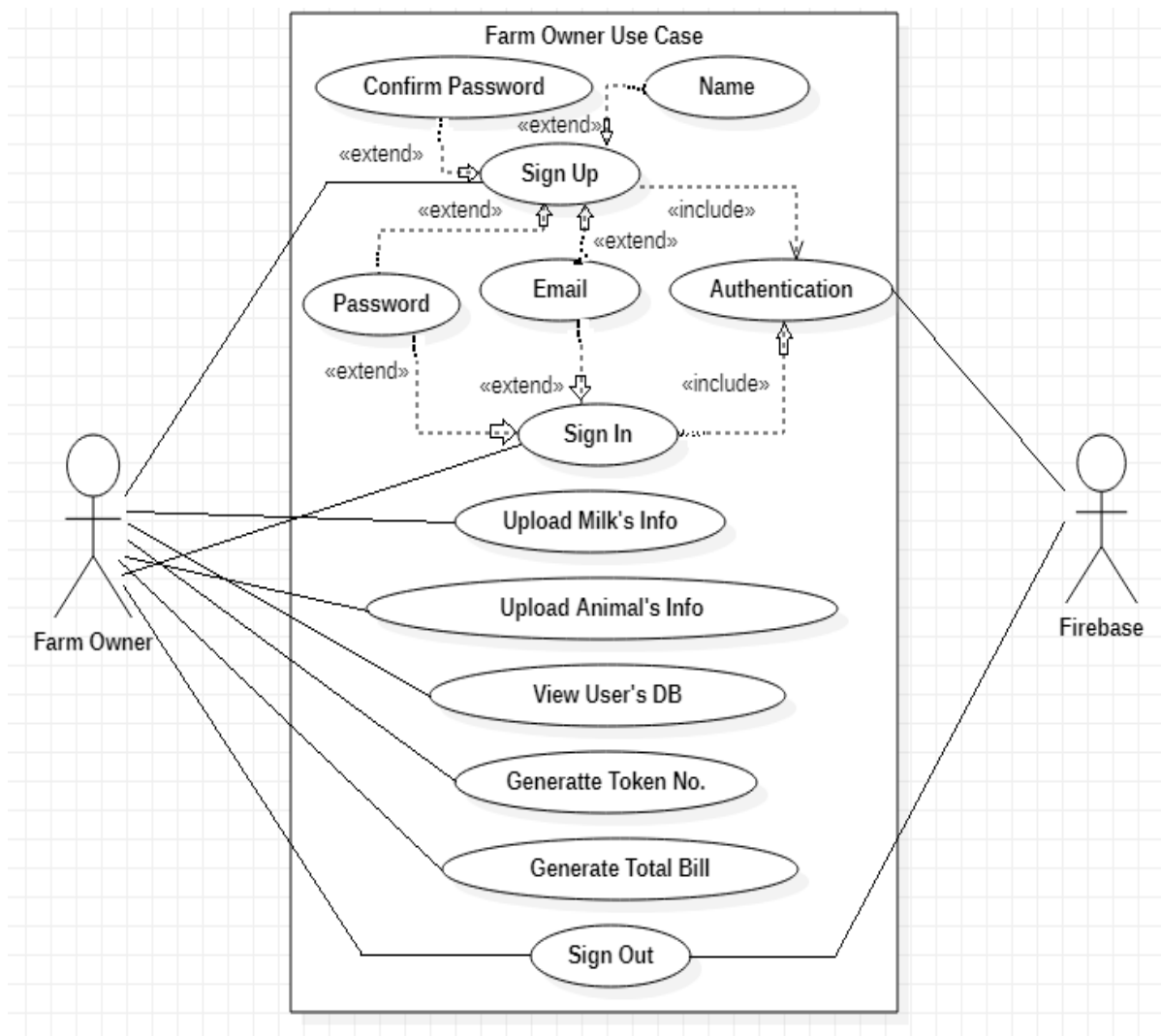


Figure: 4.2 – Farm Owner Use case Diagram

Use Case Id	UC-2
Actor	Farm Owner
Pre-Condition	Sign in/ Sign up to the system
Functionality	<ul style="list-style-type: none"> • Upload Milk's Info • Upload Animal's Info • View User's Database • Generate Token No. • Generate Total Bill
Post-Condition	Update data in a database (firebase) and sign out from the system

Table: 4.2 – Farm Owner Use case Table

4.2.3. Food Authority Use case Diagram:

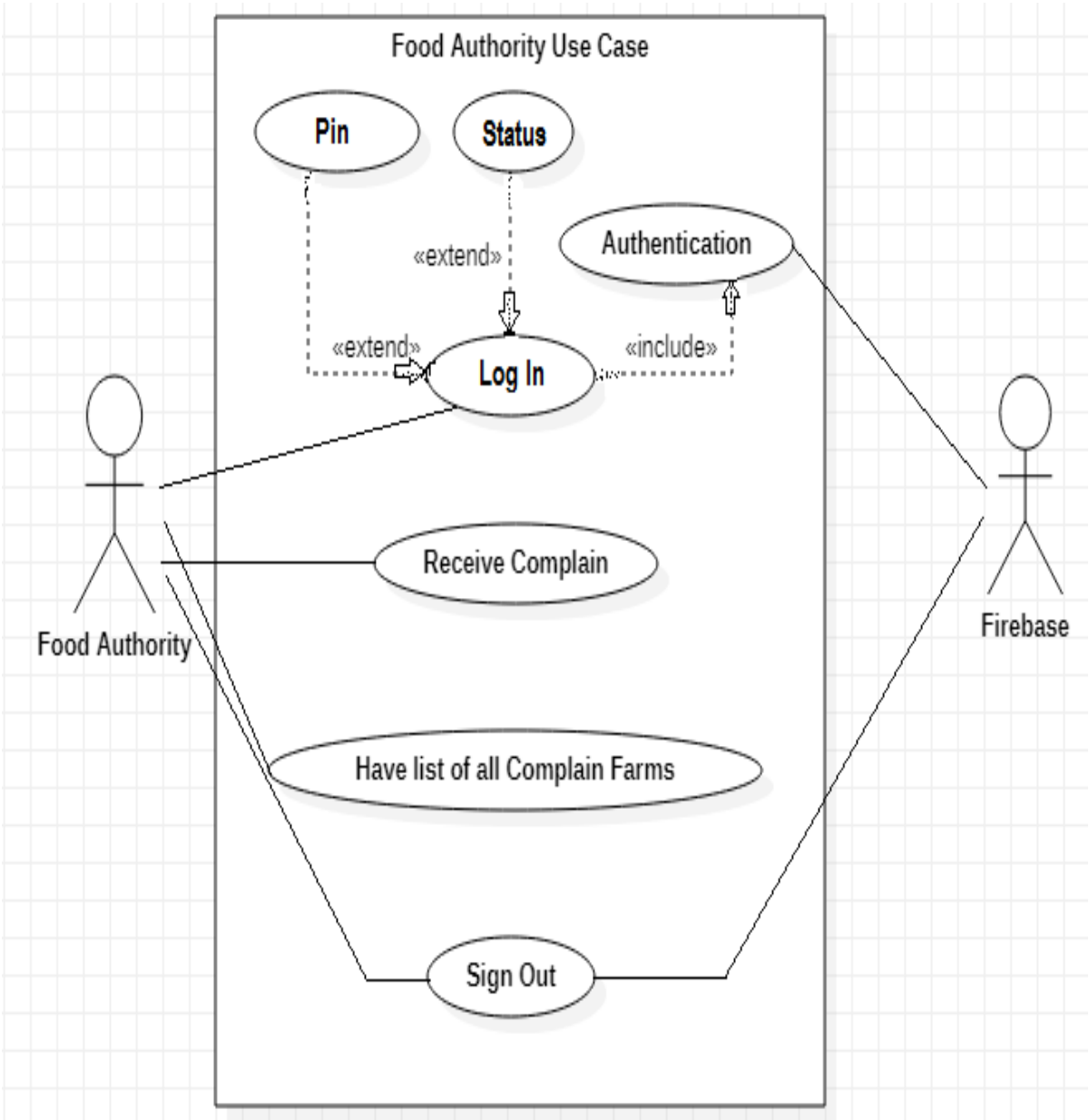


Figure: 4.3 – Food Authority Use case Diagram

Use Case Id	UC-3
Actor	Food Authority
Pre-Condition	Sign in to the system
Functionality	<ul style="list-style-type: none">• Receive Complain from the users• Maintain a list of Complained farms• Take action against particular farm
Post-Condition	Sign out from the system

Table: 4.3 – Food Authority Use case Table

4.3. Activity Diagram:

An activity diagram is used to show the flow of activities of the system. It uses different symbols for different operations like a circle for start, the circle filled with another circle for the end, parallelogram for input/output, rectangle for a simple statement, diamond for comparisons and fork and join, etc. It uses arrows to move from one activity to another.

4.3.1. Farm Owner Activity Diagram:

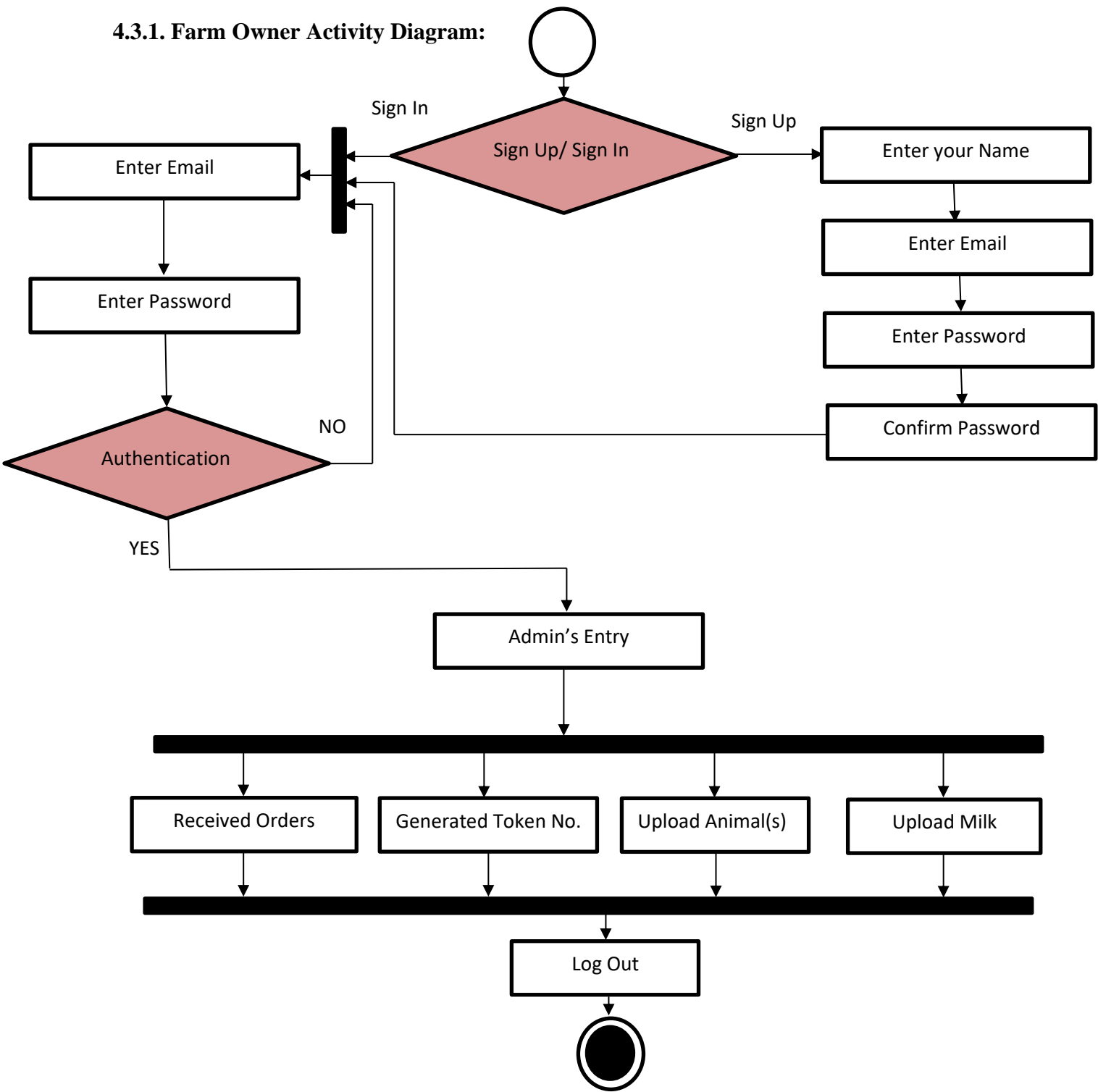


Figure: 4.4 – Farm Owner Activity Diagram

4.3.2. User Activity Diagram:

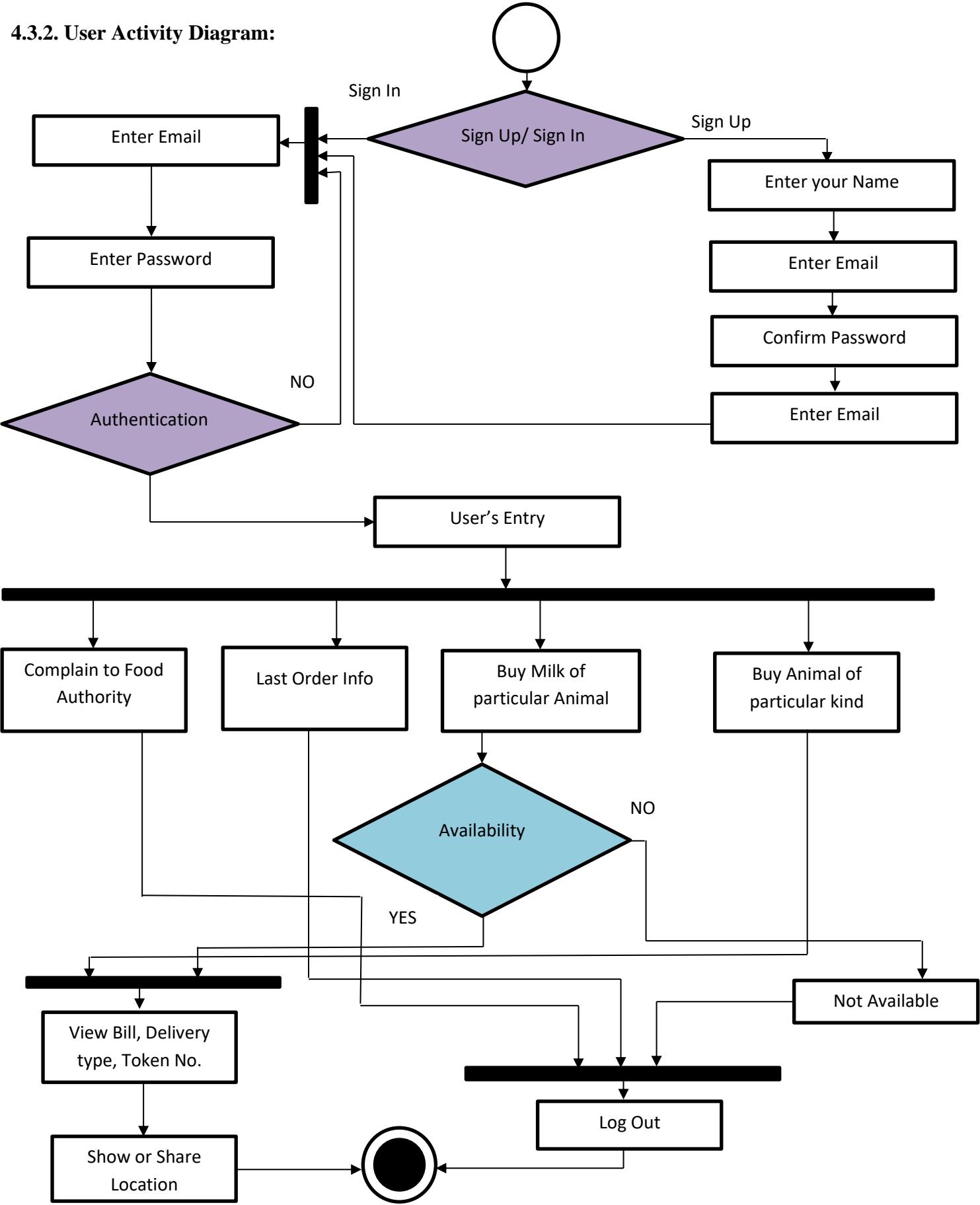


Figure: 4.5 - User Activity Diagram

4.4. Data Flow Diagram:

The data flow diagram is used to show the flow of data or information between different modules participating in the system. DFD is short term Data Flow Diagram. DFD has different levels known as 0 Level, 1 Level, 2 Level, and so on. It includes data inputs, data outputs, data stores, and various sub-processes the data move through.

4.4.1. 0-Level DFD of the System:

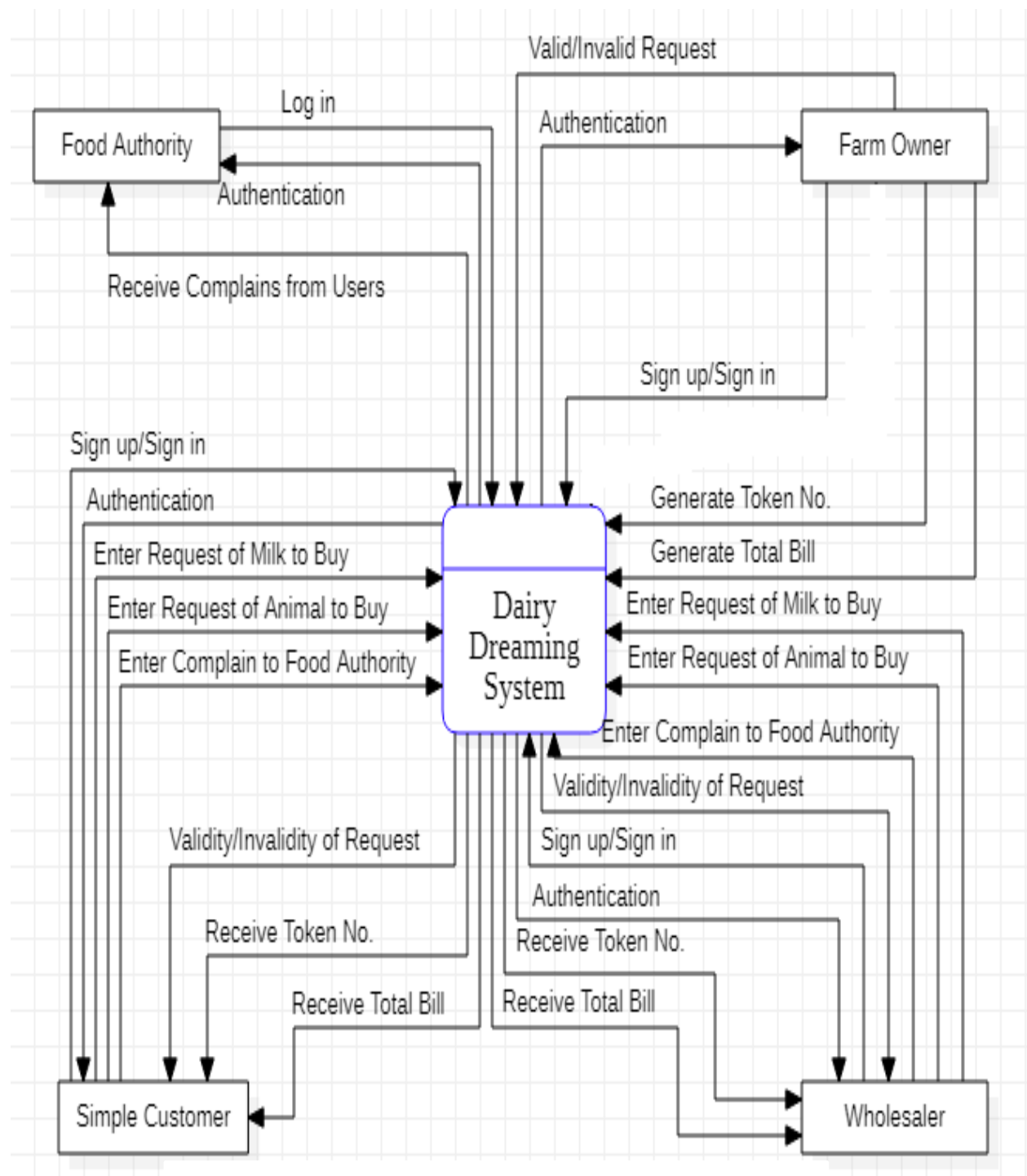


Figure: 4.6 – Data Flow Diagram (Level-0) of the System

4.4.2. 1-Level DFD of the System:

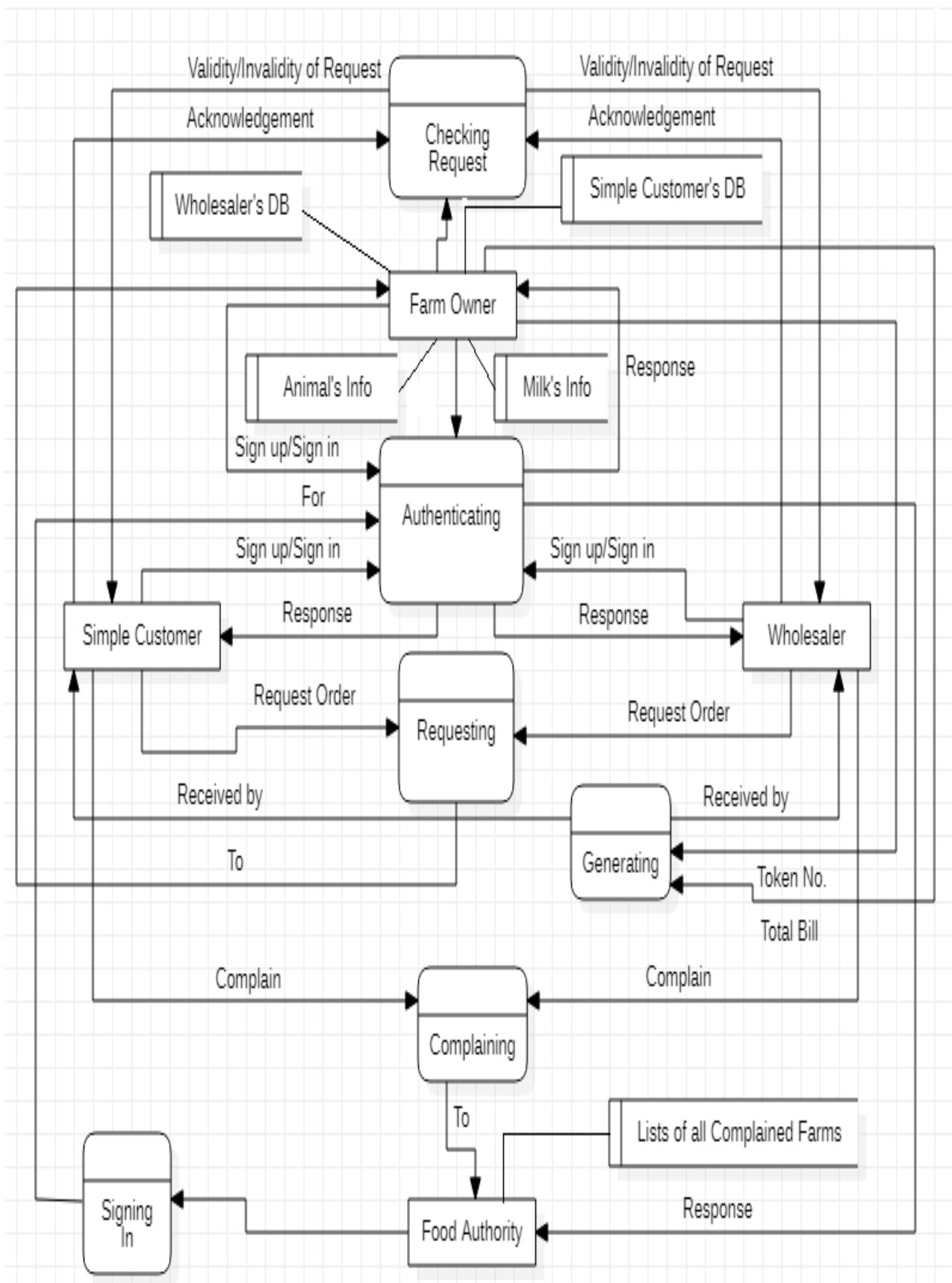


Figure: 4.7 - Data Flow Diagram (Level-1) of the System

4.5. Class Diagram:

The class diagram is used to show the attributes, operations, and all the constraints imposed by the system. The class diagram is used to show the relationship between/among all the modules of the system.

4.5.1. Class Diagram of the System:

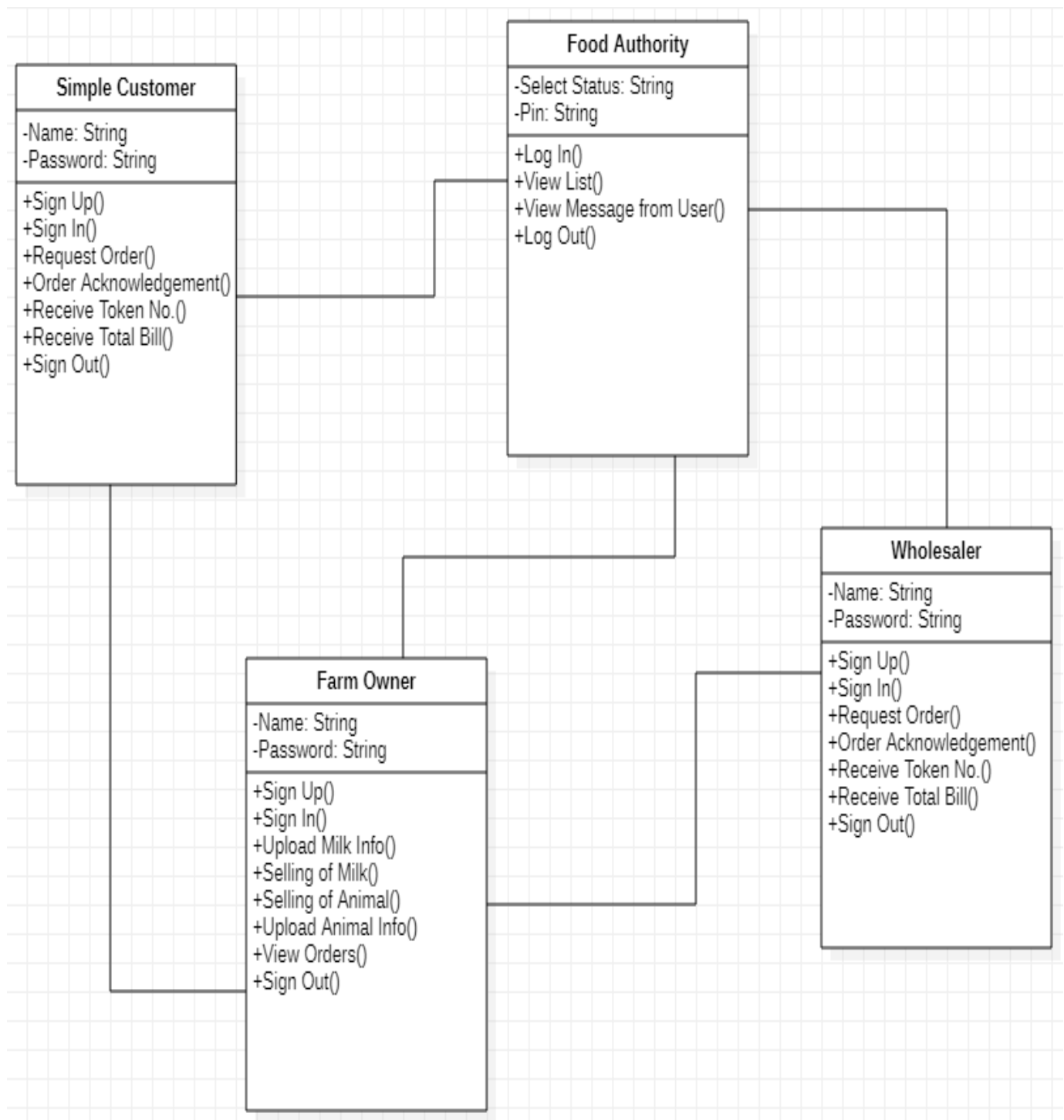


Figure: 4.8 – Class Diagram of the System

4.6. Sequence Diagram:

A sequence diagram is used to show all operations in a sequence. It describes how and in which order different objects of the system work together in close collaboration.

4.6.1. Farm Owner Sequence Diagram:

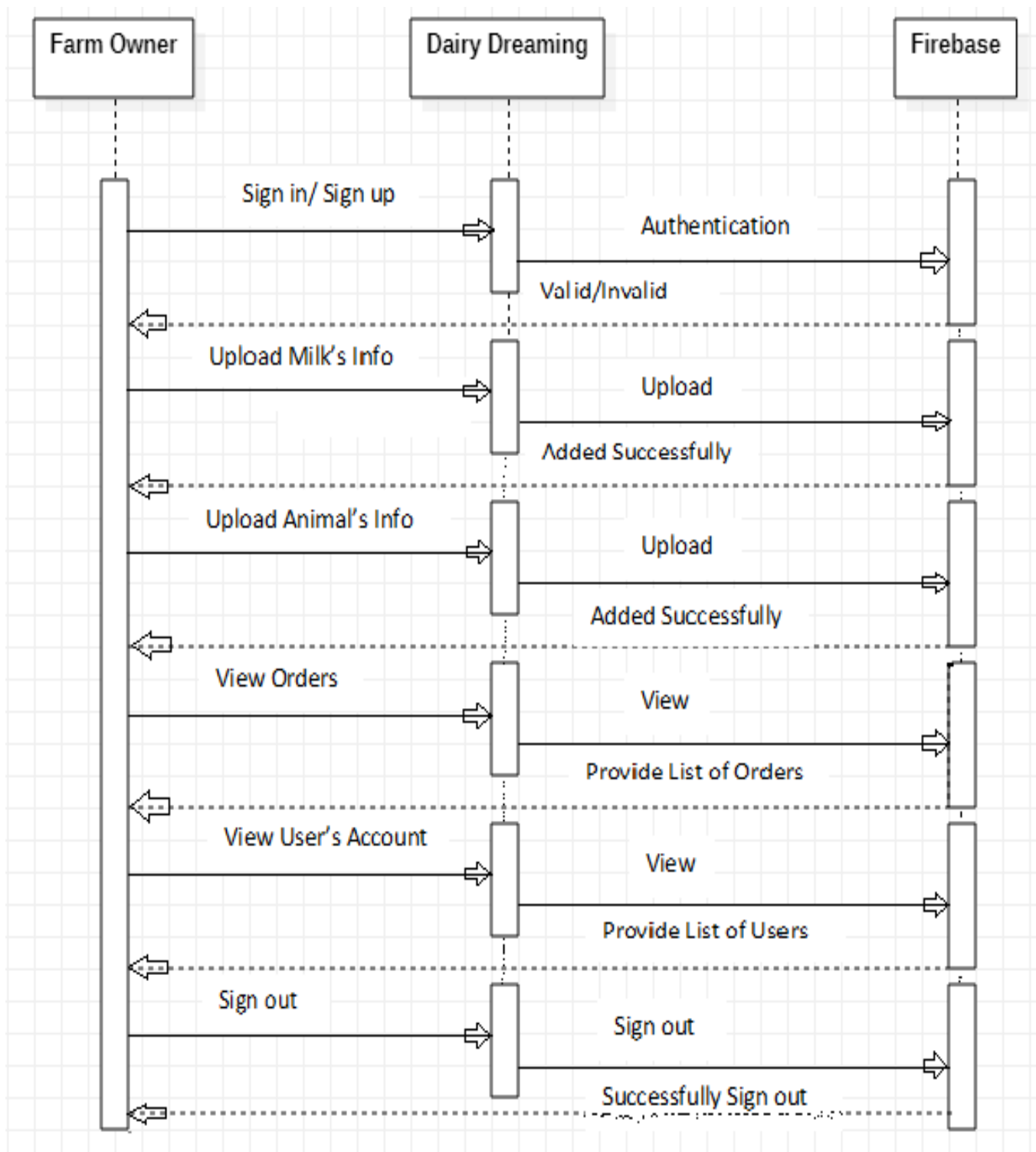


Figure: 4.9 – Farm Owner Sequence Diagram

4.6.2. User Sequence Diagram:

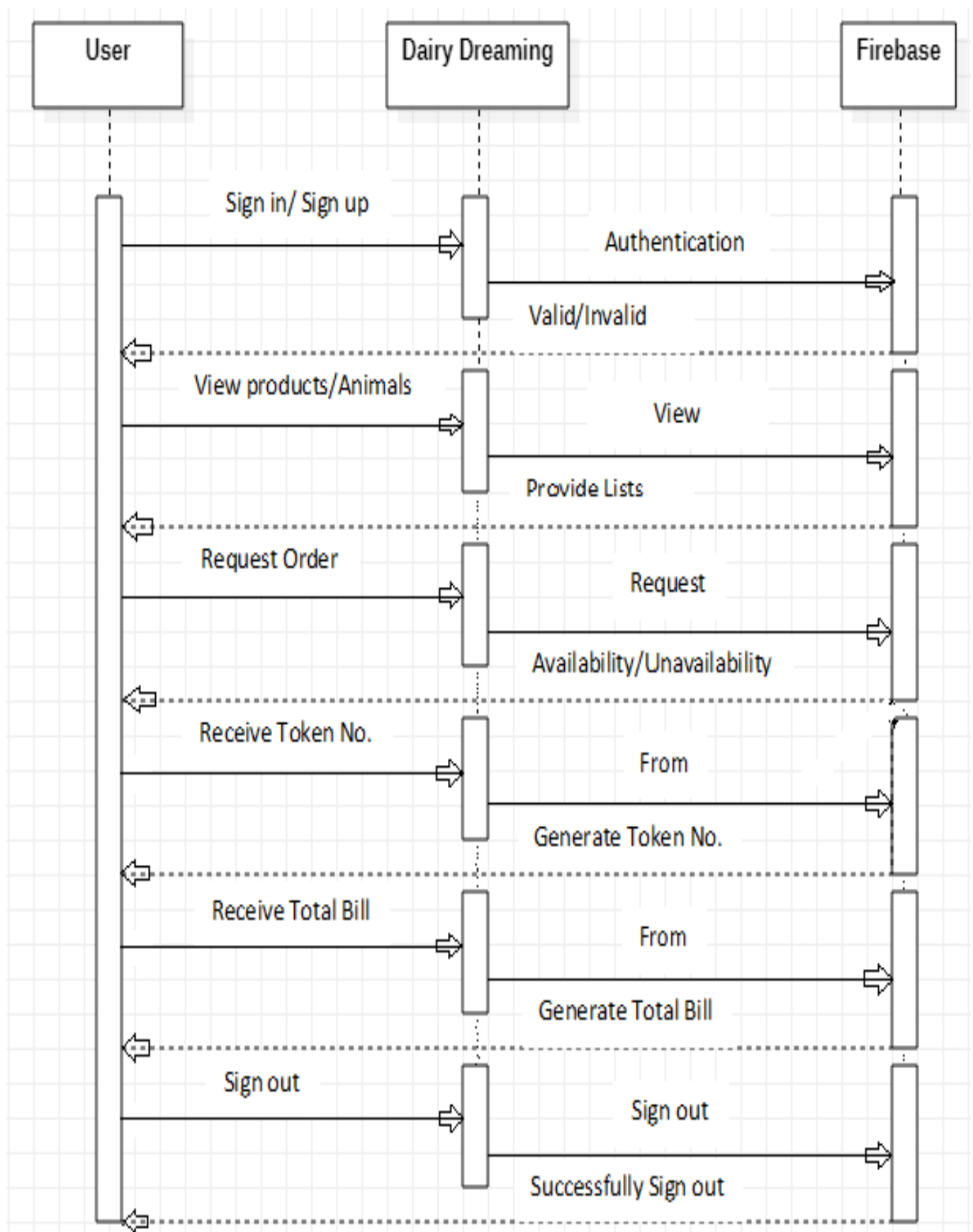


Figure: 4.10 – User Sequence Diagram

Chapter 5

Implementation

5.1. Implementation:

In this chapter, we will elaborate on the implementation of our proposed system. We will discuss the tools and technologies that we used to develop the project. In this phase, we will convert the theoretical idea of our project into the physical environment.

5.2. Tools and Technologies:

Tools are the most essential part of the development of any project. A software developer cannot write the code without these tools. Technologies are also one of the key parts of any software project. Technologies provide the basic building block in the development of any software system.

5.2.1. Tools:

- Android Studio

5.2.2. Languages:

- Java (and JavaScript)
- XML
- HTML and CSS

5.2.3. Data Access:

- Firebase
- Google Map

5.3. Software Requirement:

- MS Word (for documentation)
- MS PowerPoint (for presentation)
- Star UML (for diagrams)

5.4. Hardware Requirements:

- Intel® Core™ i5 (Laptop or PC)
- Android Smartphone or Emulator

5.5. User Interface:

The user interface of every project should be interactive and simple so that everyone can use the system easily without facing any difficulty. Below are some user interfaces (screenshots) of our project that made our project interactive and efficient.

5.5.1. Splash Screen:

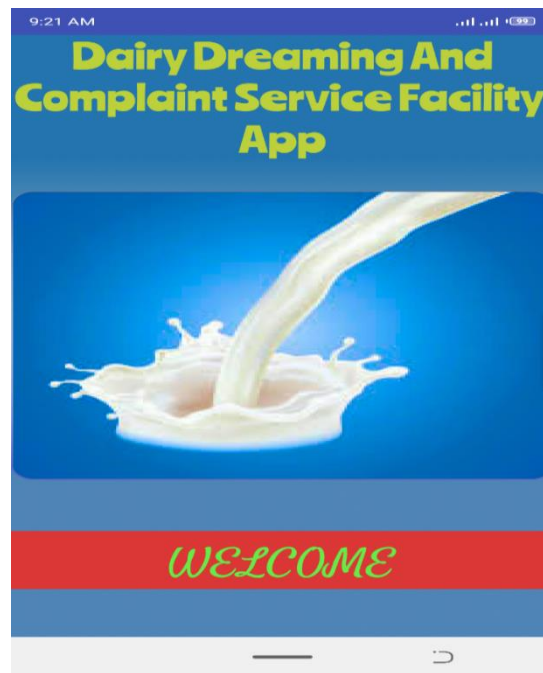


Figure: 5.1- Splash Screen

Figure 5.1 shows the splash screen of the Dairy Dreaming and Complain Service Facility App. Which will disappear after 2 sec.

5.5.2. Switch Board:

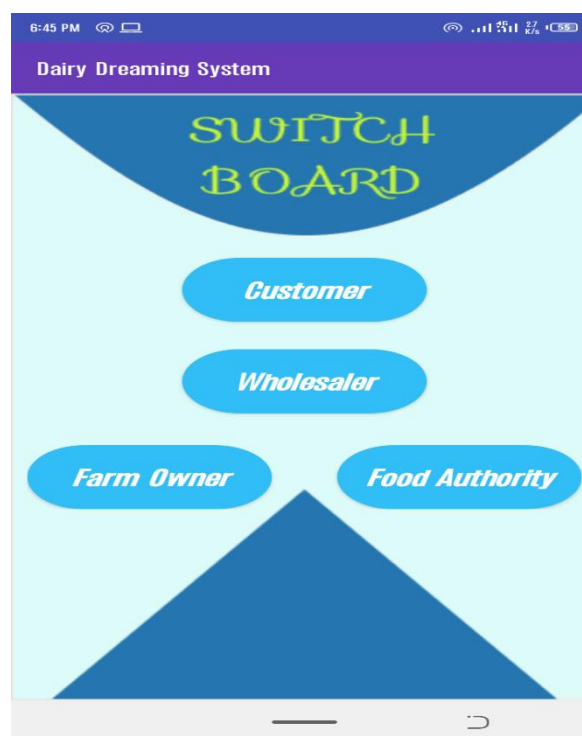


Figure: 5.2- Switch Board

Figure 5.2 shows the Switch Board. The user uses this switchboard according to his/her nature.

5.5.3. Sign Up Activities:

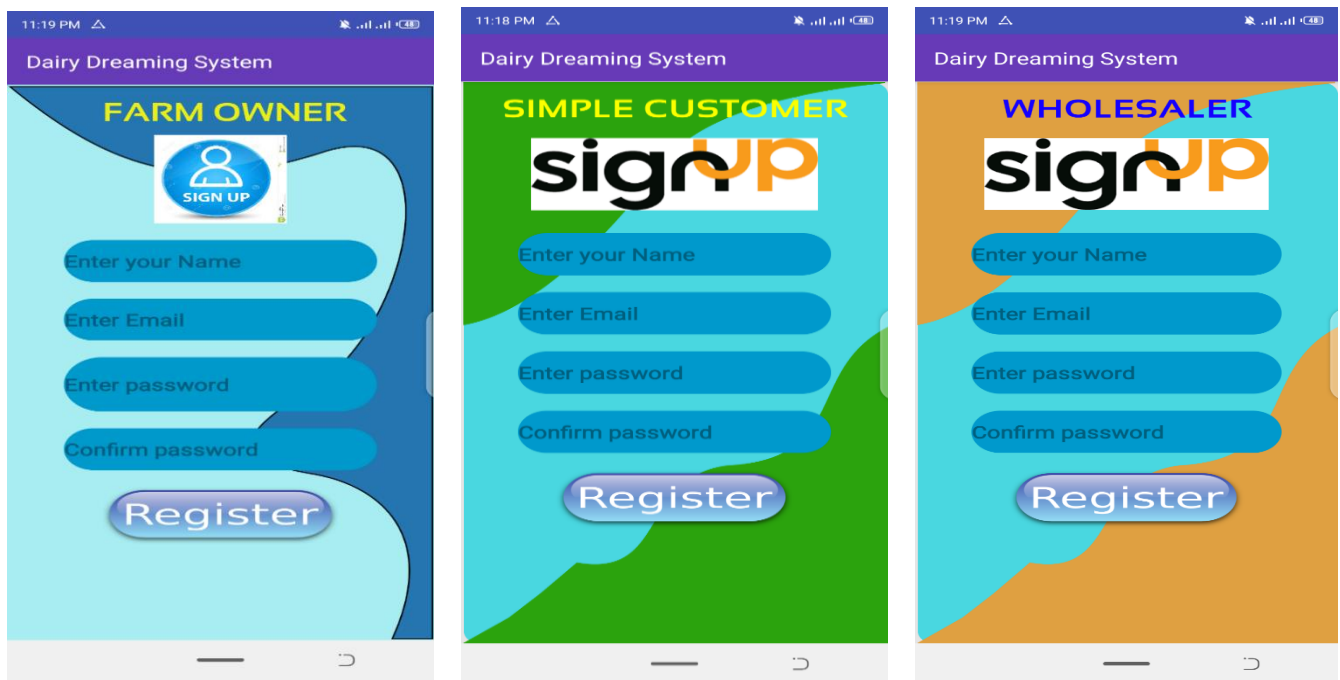


Figure: 5.3- Sign Up Activities

Figure 5.3 shows the sign-up activity of Farm Owner, Simple Customer, and Wholesaler which require Name, Email, Password and Confirm Password for registration.

5.5.4. Sign In Activities:

5.5.4.1. Sign In (1):

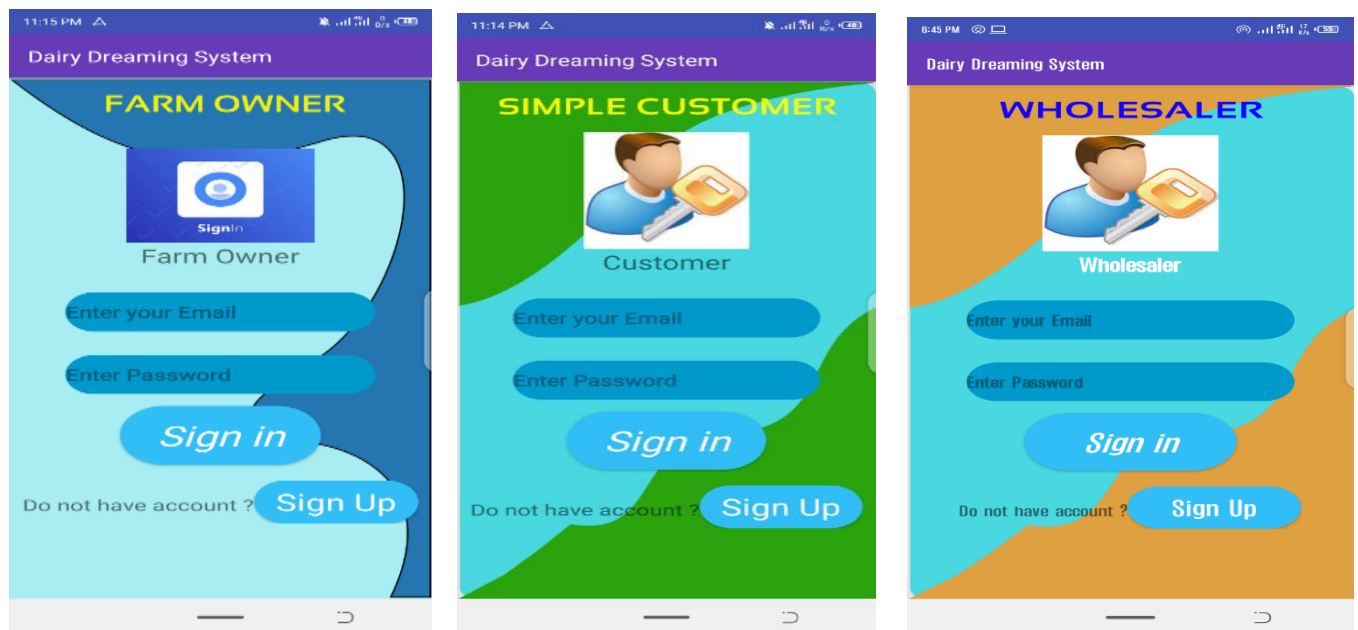


Figure: 5.4- Sign In Activities (1)

Figure 5.4 shows the sign-in activity of Farm Owner, Simple Customer, and Wholesaler which require Email and Password for signing in to the system.

5.5.4.2. Sign In (2):

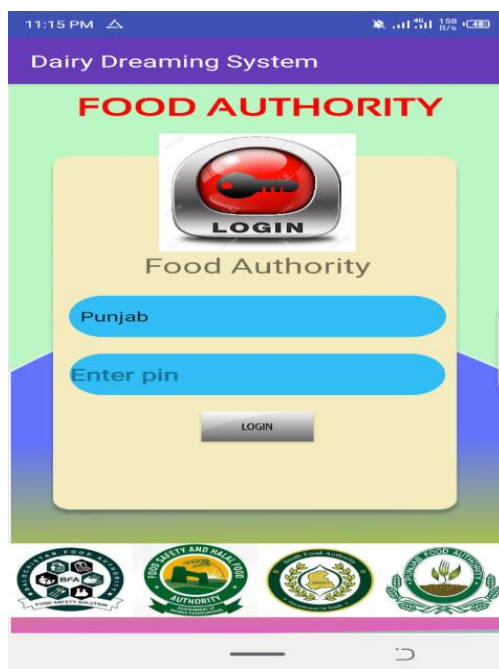


Figure: 5.5- Sign In Activity (2)

Figure 5.5 shows the sign-in activity of the Food Authority, which food authority uses Status and Pin for signing in to the system.

5.5.5. Associated Action for the User:

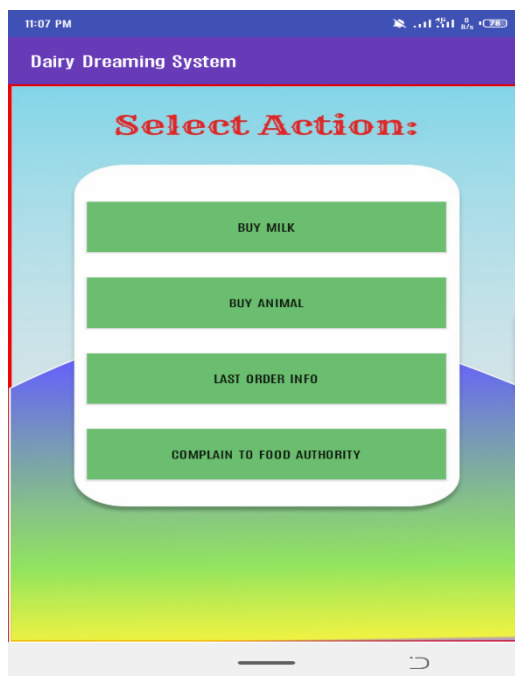


Figure: 5.6- Associated Action for the User

Figure 5.6 shows the associated action for the user i.e., Simple Customer or Wholesaler either to Buy Milk, to Buy Animal, or to complain to Food Authority.

5.5.6. List of all available Farms:

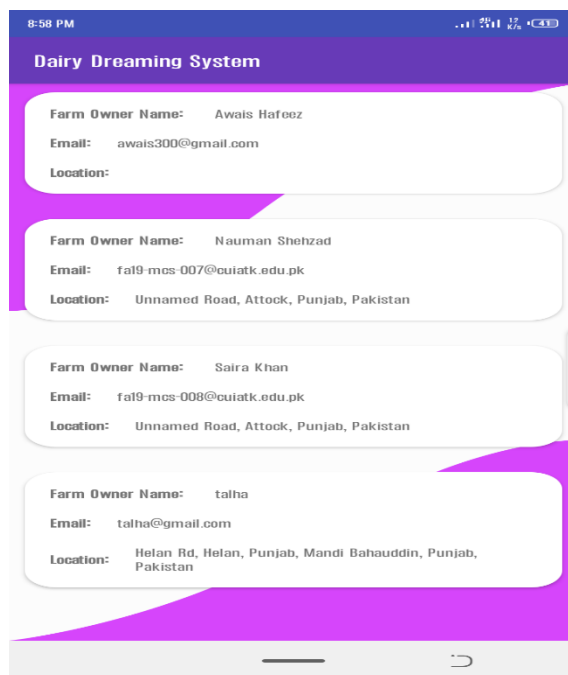


Figure: 5.7- List of all Available Farms

Figure 5.7 shows the lists of all available farms with the Name, E-mail of the farm owner.

5.5.7. Ordering of Milk via Offline or Online delivery:

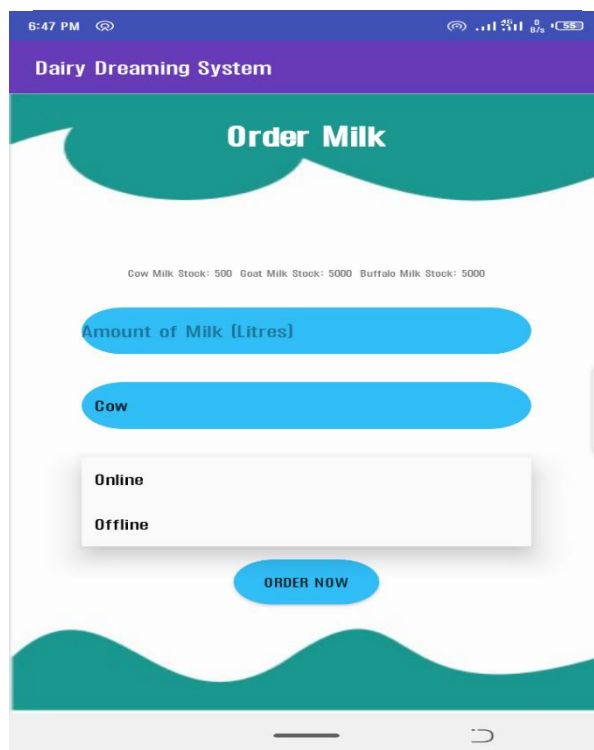


Figure: 5.8- Ordering of Milk

Figure 5.8 shows the ordering of Milk via offline or online delivery (extra recharge or Rs.100 for online delivery per Animal’).

5.5.8. Ordering of Animal via Offline or Online delivery:

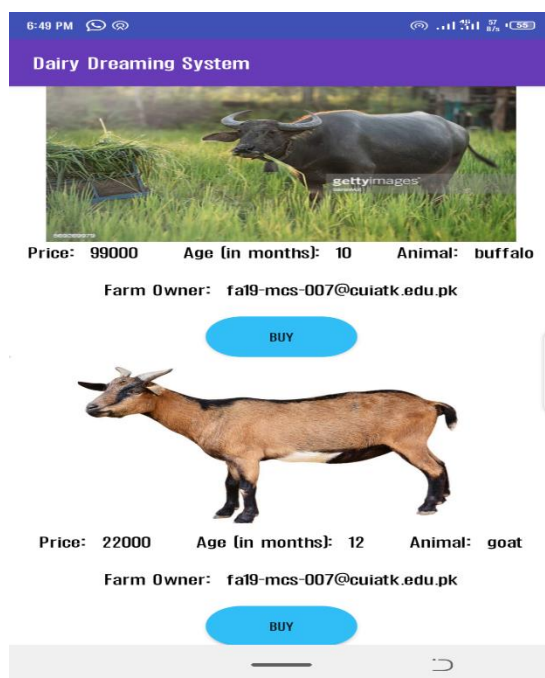


Figure: 5.9- Ordering of Animals

Figure 5.9 shows the Ordering of Animals of a particular type offline or online (with an extra charge of Rs.500 for online delivery).

5.5.9. Location (dependent on the Delivery method):

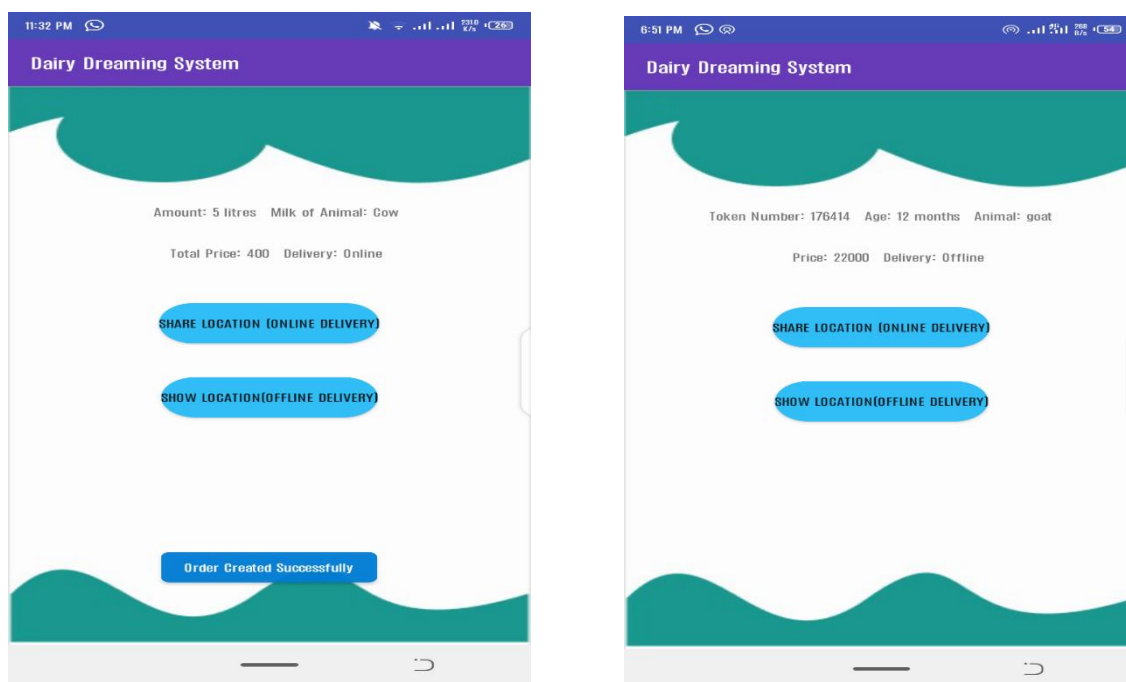


Figure: 5.10- Location Option

Figure 5.10 shows the sharing option for the user via offline or online. In case of online delivery, the location of the farm will be shown to the user and in case of offline delivery the location of that user will be shared with a Token No.

5.5.10. Locations of the Farm (in case of Online delivery):

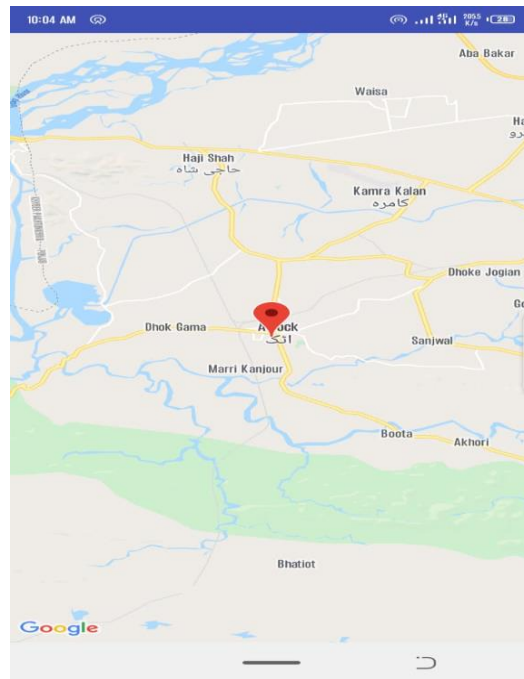


Figure: 5.11- Location of the Farm

Figure 5.11 shows the location of the Farm in case of online delivery.

5.5.11. Location of the User (in case of Offline delivery):

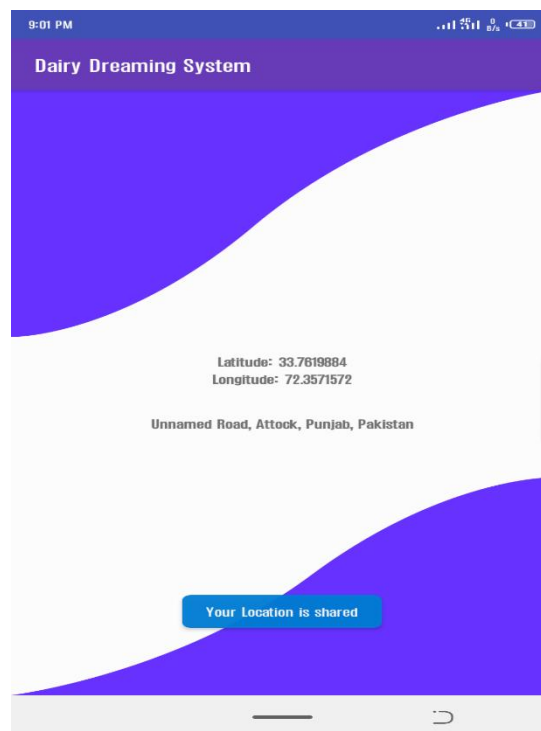


Figure: 5.12- Location of the User

Figure 5.12 shows the location of the User in case of offline delivery (Token No. will be generated to the user).

5.5.12. Associated action for Farm Owner:

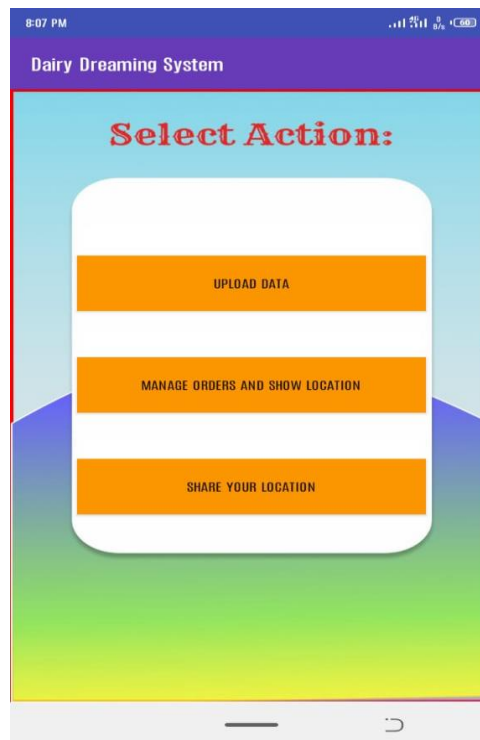


Figure: 5.13- Associated Action for Farm Owner

Figure 5.13 shows the associated action for the farm owners.

5.5.13. Upload Data:



Figure: 5.14- Upload Data

Figure 5.14 shows the uploading of Milk or Animal by Farm Owner

5.5.14. Update or Upload Milk:



Figure: 5.15- Update Price and Stock

Figure 5.15 shows the updating or uploading of Milk price and stock by the farm owner.

5.5.14.1. Update or Upload Price of each Milk type:

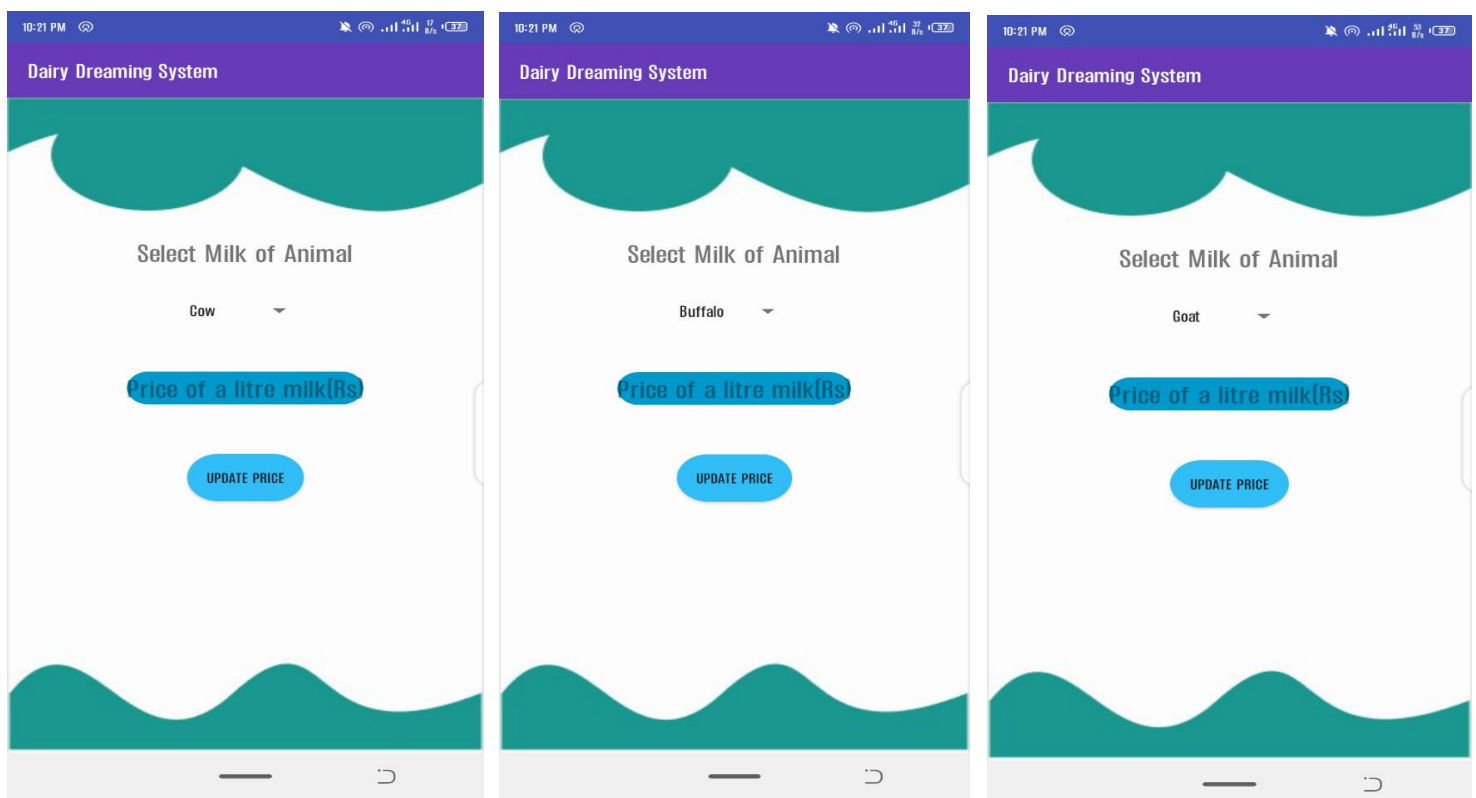


Figure: 5.16- Update, Upload Price of Milk

Figure 5.16 shows the updating or uploading of the price of Milk of each Animal.

5.5.14.2. Update or Upload Stock of each Milk type:

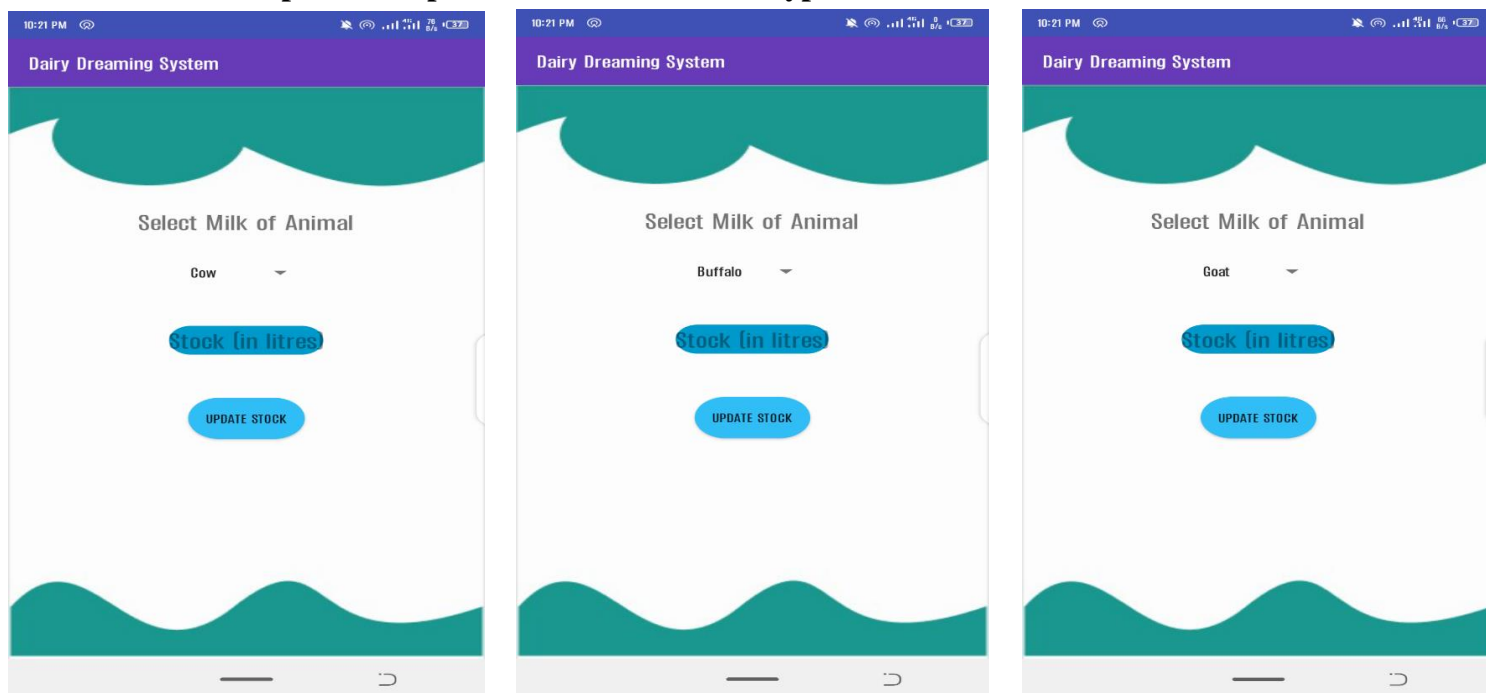


Figure: 5.17- Update, Upload Stock of each Milk type

Figure 5.17 shows the updating of the stock of Milk.

5.5.15. Upload Animal of each type:



Figure: 5.18- Upload Animal

Figure 5.18 shows the Uploading of each Animal type by the farm owner.

5.5.16. Upload Animal info:

The figure displays three mobile application screens for uploading animal information. Each screen has a purple header with the text 'Dairy Dreaming System'. The first screen is titled 'Cow Data', the second 'Buffalo Data', and the third 'Goat Data'. Each screen contains two input fields labeled 'Age (in months)' and 'Price', a large grey rectangular area for an image, and a blue button labeled 'UPLOAD DATA' at the bottom. The screens are decorated with green wavy patterns at the top and bottom.

Figure: 5.19- Upload Animals' Info

Figure 5.19 shows the uploading Info of each Animal by Farm Owner.

5.5.17. Manage orders of Milk of the Users:

The figure shows a mobile application screen titled 'Dairy Dreaming System' with a search icon. It displays a list of milk orders. Each order card contains the following information:

- Customer Email: fa19-mcs-222@cuiatk.edu.pk
- Milk of Animal: Cow
- Amount: 2
- Total Price: 300
- Delivery: Online

The second order card shows:

- Customer Email: fa19-mcs-777@cuiatk.edu.pk
- Milk of Animal: Buffalo
- Amount: 1
- Total Price: 210
- Delivery: Online

The third order card shows:

- Customer Email: fa19-mcs-777@cuiatk.edu.pk
- Milk of Animal: Cow
- Amount: 1
- Total Price: 90
- Delivery: Offline

The bottom of the screen shows a partially visible order card with the label 'Customer Email:'.

Figure: 5.20- Manage Orders of Milk

Figure 5.20 shows all the orders of Milk managed by the Farm Owner.

5.5.18. Manage orders of Animals of the Users:

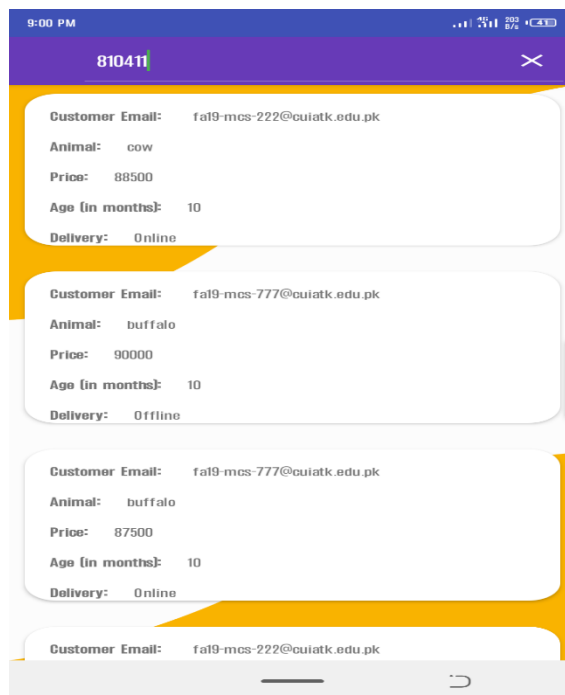


Figure: 5.21- Manage Orders of Animals

Figure 5.21 shows all the orders of Animals managed by the Farm Owner.

5.5.19. List of Farms that are complained by the User:

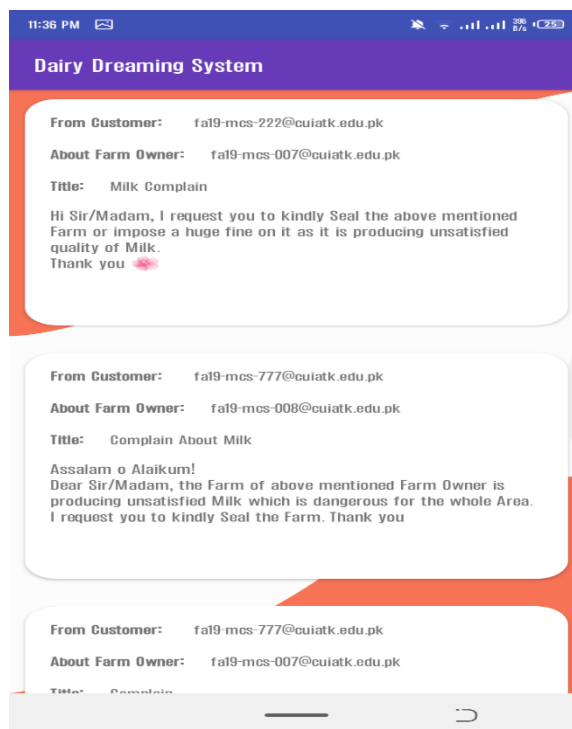


Figure: 5.22- Lists of all Complained Farms

Figure 5.22 shows the lists of all farms that are complained by the User.

5.5.20. Complain Form for the User:

The screenshot shows a mobile app interface for the 'Dairy Dreaming System'. The title bar is purple with the text 'Dairy Dreaming System'. Below it is a green header with the title 'Complain Form'. The form consists of three blue rounded rectangular input fields: 'Title of Complain', 'Farm Owner Email', and 'Punjab'. Below these is a large yellow rectangular text area with the placeholder text 'Enter your message here:'. At the bottom of the form is a blue rounded rectangular button labeled 'COMPLAIN'. The app is running on a device with a status bar at the top showing the time as 6:59 PM and various icons.

Figure: 5.23- Complain Form for the User

Figure 5.23 shows the Complain Form for the user to Food Authority.

5.5.21. Milk Data of Selected Farm:

The screenshot shows a mobile app interface for the 'Dairy Dreaming System'. The title bar is purple with the text 'Dairy Dreaming System'. Below it is a green header with the title 'Milk data of Selected Farm'. The data is displayed in three rows, each with a label and a value: 'Cow Milk Price : 90' and 'Cow Milk Stock : 9000', 'Buffalo Milk Price : 120' and 'Buffalo Milk Stock : 5000', and 'Goat Milk Price : 80' and 'Goat Milk Stock : 5000'. The app is running on a device with a status bar at the top showing the time as 11:50 PM and various icons.

Figure: 5.24- Milk Data of Selected Farm

Figure 5.24 shows the Milk data of the selected Farm.

Chapter 6

Evaluation

6.1. Evaluation:

In this chapter, we will evaluate our proposed system by applying different testing strategies to check the normal working of the application.

6.2. Testing:

Testing is a very important phase in the development of any application. Testing is the process of analyzing the proposed system to check whether the system meets all its functional and nonfunctional requirements or not. It is also used to detect bugs (errors) in the system if exist. Testing should be addressed continuously throughout the development of the app rather than at the end. To evaluate our system, we will use unit testing, integration testing, and system testing.

6.2.1. Unit Testing:

In unit testing, each module of the system is tested individually. After applying unit testing, you must ensure that all the modules are working well with any deficiency or problem.

6.2.2. Integration Testing:

In integration testing, distinct software modules are collected and tested in the form of a group. It is applied after the unit testing. Integration testing walks as input each module in unit testing, batches them, and provides its output to be prepared for system testing.

6.2.3. System Testing:

In system testing, the entire system is evaluated as a whole to assess the system's agreement with its agreed requirements. System testing is a more restricted type of testing; it pursues to detect defects both within the "inter-assemblages" and also within the entire system.

6.3. Test Cases:

The test cases are executed during the project development. They are used to test the functional and non-functional requirements of the project.

6.3.1. Test Case 1 (Sign Up):

Test #	Action	Expected Result	Actual Result	Status
1	Click on the Register button without entering any data	An error message will occur in field as Name, Password, Email cannot be empty	A message will display for a required field(s) as it cannot be empty	Pass
2	Click invalid Email and click on the Sign-Up button.	An error message will occur in the field as entered Email is invalid	A message will display for the required field as the Email is not valid.	Pass
3	Enter valid Name, Email, Password, and Confirm Password.	The app will successfully Sign Up the user.	Successfully Signed Up.	Pass

Table: 6.1 – Test Case 1 (Sign Up)

6.3.2. Test Case 2 (Sign In):

Test #	Action	Expected Result	Actual Result	Status
1	Click on the Sign In button without entering any data	An error message will occur in the field as Email cannot be empty.	A message will display for the required field(s) as it cannot be empty.	Pass
2	Click invalid Email and click on the Sign In button.	An error message will occur in the field as the Email is invalid.	A message will display as an Email in the not valid.	Pass
3	Enter valid Email and Password	The app will successfully Sign In to the user.	Successfully Signed In.	Pass

Table: 6.2 – Test Case 2 (Sign In)

6.3.3. Test Case 3 (Ordering of Milk):

Test #	Action	Expected Result	Actual Result	Status
1	Click on Order Now button without entering Amount of Milk (Liters)	An error message will occur in the field as must enter the amount of Milk.	A message will display for the required field(s) as it cannot be empty.	Pass
2	Enter Invalid amounts of invalid data of Milk (Liters).	The app's functionality will not work properly.	The app has stopped it's working.	Pass
3	Enter correct and valid Password.	The order will be created for the user.	Order Created.	Pass

Table: 6.3 – Test Case 3 (Ordering of Milk)

6.3.4. Test Case 4 (Uploading of Milk):

Test #	Action	Expected Result	Actual Result	Status
1	Click on the Update Price and Update Stock button without entering any input.	An error message will occur in fields as. Price and Stock cannot be empty.	A message will occur in the required fields as "Price cannot be empty" and "Stock cannot be empty".	Pass
2	Enter correct Price and Stock in the respective fields.	Price and Stock of a particular kind will be updated.	Price Updated and Stock Updated	Pass

Table: 6.4 – Test Case 4 (Uploading of Milk)

6.3.5. Test Case 5 (Uploading of Animal):

Test #	Action	Expected Result	Actual Result	Status
1	Click on the Upload Data button without entering any input.	An error message will be displayed for the required fields.	The app has stopped it's working.	Pass
2	Enter Invalid Age, Price or Image of a particular animal	An error message will be displayed for the required fields.	The app has stopped it's working.	Pass
3	Enter correct Age, Price, and Image.	Animal of a particular kind will be uploaded.	Image Uploaded	Pass

Table: 6.5 – Test Case 5 (Uploading of Animal)

Chapter 7

Conclusion and Future Work

7.1. Conclusion and Future Work:

In this chapter, we will discuss the conclusion and future work of our project after the completion of the Implementation and Evaluation phases.

7.1.1. Conclusion:

We have developed our android application in full working condition having expected functionalities and outcomes. We have achieved our target goal by utilizing a minimum number of resources. “Dairy Dreaming and Complain Service Facility App” is a responsive android application that can be easily accessed by the user in his/her smartphone with a proper internet connection. It was our primary goal to develop an app with efficient performance and outcomes. Our project is about selling Milk and Animals of a particular kind. The separates rates for Wholesalers and Simple Customers are shown when they sign in/up to the app. Any user can create an account on the system without any restriction according to his/her nature and job just by using his/her smartphone. The Simple Customer and Wholesaler can buy Milk and Animal of different kinds as well as can complain to Food Authority for the bad standard and quality of the product. All the info on Milk and Animal is uploaded by the Farm Owner of a particular farm. The farm owner will have a list of all the users along with the delivery type that requests an order of that farm.

Both the farm owner and the user can find each other easily with the help of location provided via Google Map. This application is available at mobile application stores or similar services so that users can download and install it without any cost. The system will authenticate all the users involved in the application.

7.1.2. Future Work:

Now our application is about the selling of Milk and animals. In the future, we can improve our application in several ways by adding and selling more products like yogurt, butter, etc. In the future, we can add different distributors to our app for the effective delivery of the products. Also, we can add different experts that can guide us about the products. We can add different payment methods such as ATM card, Easypaisa, Jazz cash, etc. so that the user can use digital methods to deposit his/her bill to the farm.

Chapter 8

References

8.1 References:

- [1] <https://studentprojectguide.com/php/milk-dairy-management-system/amp/>
- [2] https://en.wikipedia.org/wiki/Android_software_development
- [3] <https://www.youtu.be/EplH-amHTtE>
- [4] <https://www.youtu.be/qK0QNA0sMGc>
- [5] <https://abhiandroid.com/programming/googlemaps>
- [6] <https://firebase.google.com/docs/database/android/read-and-write>
- [7] <https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/dairy-products>