

A Project Report On
DAIRY MANAGEMENT SYSTEM

Submitted by
MOHAMMED MUSTHAFA V
(MMAVMCA026)

Guided by
Mrs. SURYA U.
(Faculty, CCSIT Mannarkkad)

Submitted in partial fulfillment of the requirements for the Award
of the degree of
MASTER OF COMPUTER APPLICATIONS



UNIVERSITY OF CALICUT
CENTRE FOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
MANNARKKAD
2021-2023

**CENTRE FOR COMPUTER SCIENCE AND INFORMATION
TECHNOLOGY MANNARKKAD**

2021-2023



UNIVERSITY OF CALICUT

CERTIFICATE

This is to certify that the project entitled as **‘DAIRY MANAGEMENT SYSTEM’** is a bonafide record of original work done by **MOHAMMED MUSTHAFA V (MMAVMCA026)** towards the partial fulfillment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** under **UNIVERSITY OF CALICUT**. We also certify that the work is original and has not been submitted to any other university wholly or in a part of any degree or diploma.

Mrs. PRIYA TR

Centre in charge
CCSIT Mannarkkad

Mrs. SURYA U

Internal Guide
CCSIT Mannarkkad

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the project entitled as '**DAIRY MANAGEMENT SYSTEM** ' submitted to University of Calicut in a partial fulfillment of the requirement for the award of the degree of Master of Computer Applications, is a record of original work done by me, during the period of study at CCSIT Mannarkkad under the supervision and guidance of **Mrs. SURYA U**, faculty of CCSIT Mannarkkad.

Place: Mannarkkad

MOHAMMED MUSTHAFA V

Date:

(MMAVMCA026)

ACKNOWLEDGEMENT

I take this opportunity to express my heartiest gratitude to all, whose contribution in this project work can never be forgotten.

I express my heartfelt thanks to the guide and staffs of RISS TECHNOLOGIES for their full-hearted help and encouragement throughout this project.

I am extremely grateful to my beloved teachers especially my internal guide Mrs. SURYA U and our centre in charge Mrs. PRIYA T.R, for their effort and numerous suggestions and also for their kind co-operation throughout this project.

Last but not least I would like to thank my beloved parents and friends who showered up on me their love and constant encouragement for completion of this project work and believed in my ability.

Once again, I thank to all who has helped me directly and indirectly for the successful completion of the project work.

.

MOHAMMED MUSTHAFA V

(MMAVMCA026)

TABLE OF CONTENTS

CONTENTS	PAGE NO
1. INTRODUCTION	1
1.1 SYNOPSIS	2
1.2 ORGANIZATION PROFILE	3
2. SYSTEM ANALYSIS AND SYSTEM STUDY	4
2.1 SYSTEM ANALYSIS	5
2.2 SYSTEM STUDY	8
2.3 EXISTING SYSTEM	9
2.4 PROPOSED SYSTEM	10
2.5 FEASIBILITY STUDY	11
2.6 SYSTEM ENVIRONMENT	14
2.7 FRONT END	15
2.8 BACK END	24
3. SYSTEM DESIGN	26
3.1 SYSTEM DESIGN	27
3.2 INPUT DESIGN	29
3.3 OUTPUT DESIGN	30
3.4 DATAFLOW DIAGRAM	31
3.5 DATABASE DESIGN	42

4. SYSTEM DEVELOPMENT	55
4.1 MODULE DESCRIPTION	56
5. SYSTEM TESTING	59
5.1 SYSTEM TESTING	60
5.2 LEVEL OF TESTING	61
6. SYSTEM IMPLEMENTATION	63
6.1 SYSTEM IMPLEMENTATION	64
6.2 DOCUMENTATION	67
7. SYSTEM MAINTANENCE AND SECURITY	68
7.1 SYSTEM MAINTANENCE	69
7.2 SYSTEM SECURITY	71
7.3 SCOPE FOR FUTURE ENHANCEMENT	74
8. CONCLUSION	76
8.1 CONCLUSION	77
9. APPENDIX	78
9.1 SCREENSHOTS	79
9.2 SAMPLE CODE	95
10. BIBLIOGRAGHY	99
10.1 BIBLIOGRAGHY	100

INTRODUCTION

1.1 SYNOPSIS

The Dairy Management System application which deals with maintaining the inventory details of dairy product like Milk, Ghee, Curd, and Paneer. The Admin, shop, and employee are website. User, delivery boy, and farmer modules are android application part. The owners of the dairy shop can solely depend on this application. Many of us want the work to be done at a faster rate. So this application will help in reducing the time for purchasing of the products.

Even this application can help the small enterprises also. The user interface must be simple and easy to understand. This application will help in storing the records like the dairy products, employee records, and customer records, purchased information in an easy and well organized manner. This will be one of the projects that will help the owners of the dairy shop to maintain the inventory and the billing process with great ease. The aim of the project is to computerize the operations of collecting the sales details from the dealer and to book order for their next dispatching.

The modules are in here Admin, User, Shop, Farmer, Employees, Delivery boy. The admin, shop, and employee modules are websites and delivery boy, farmer, user modules are android application. They are involved in this system. Delivery boy deliver the product from admin to user and shop. Shops buy the bulk load of the product and sell to the other peoples. Farmers provide milk to admin and sell their own products to users. .Employees are the workers of the Dairy management project. Here farmer and user can interact with the chat. Delivery boy can also chat with admin. This project basically targets milk farmers of the rural areas.

1.2 ORGANIZATION PROFILE

RISS technologies, an ISO 9001: 2008 certified company focuses on transforming and running business processes and operations including those that are complex and industry-specific. Our loom is distinctive: through an unbiased, agile combination of smarter process science, targeted technology and advanced analytics, we help our clients become more competitive by making their enterprises more intelligent: adaptive, innovative, globally effective, and connected to their own clients.

We provide end-to-end expert IT solutions across entire software delivery cycle. RISS cater to every single need of our clients and help them in accomplishing their goals and objectives. RISS has significant expertise accumulated over these 15 years of specialized work with hundreds of enterprises, and we remain loyal to our heritage of operational excellence as an extension of our client's business - reflected by the best client satisfaction scores in the industry.

MISSION

Our mission is to become a market leader by consistently exceeding our customer's expectations; providing them with best breed of business solutions and deploy Information Technology potentials innovatively for the Clients' unique business & competitive advantages.

VISION

To see our self India's most celebrated & 200- company status with unique value-added contributions in information technology and services by 2020.

Guide Name :Mr Naveen john Thomas

Qualification: M Tech CS

Email :riitnaveen@gmail.com

Contact No :9526848922

Experience :9 + years

**SYSTEM ANALYSIS
AND
SYSTEM STUDY**

2.1 SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. Analysis is a detailed study of various operations performed by a system and their relationship within and outside of the system. This involves gathering information and using structured tools for analysis. System analysis is the way of studying a system with an eye on solving its problem-using computer. To analyze a system, one has to study the working of the system in detail. The system analyst has to understand the functioning and concept of the system in detail, before designing the appropriate computer- based system that will meet all the requirements of the existing system.

The various techniques used in the study of the present system are:

- Observation
- Interviews
- Research
- Discussion

PRELIMINARY INVESTIGATION

Preliminary investigation checks whether a system is developed by means of SDLC, a prototyping strategy or structured analysis method or combination of these methods. A project request should be first reviewed. The entire proposal for the required project is submitted to the selection committee for the evaluation to identify that how much this project is relevant to today's internet technology.

FACTS AND FINDING TECHNIQUES

There are several methods for gathering the sort of information. We can use all of these methods for gathering information from the users of existing system. We can introduce several fact finding techniques.

1. Research

Another fact-finding technique is to thoroughly research the problem domains. Most problems are not unique. A through research has been conducted for the functionalities to be provided by the proposed system through similar websites, journals and reference books. Information from various websites and available documents from the internet are collected for studying the proposed system problems and possible solutions.

2. Observation

Observation is fact finding technique where in the system analyst either participates in or watches a person perform activities to learn about the system. This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end users. This is an effective data collection technique for obtaining an understanding of a system.

3. Questionnaires

Another fact-finding technique is to conduct surveys through questionnaires. Requirements are collected through questionnaires and modified for the proposed system through valuable suggestions from possible end users such as developers, website handlers etc. These are special documents that allow the analyst to collect the information and opinions from respondents. The document can be mass- produced and distributed to respondent. This allows analyst to collect facts from a large number of people while maintaining uniform responses.

4. Interviews

These are fact finding techniques where by the system analyst collects information from individuals through face-to-face interaction. The personal interview is generally recognized as the most important and most often facts finding techniques. Interviews are done with developers for finding their requirements needed in the system about their required interfaces and suggestions are used to develop the system interfaces and functionalities and its feasibility is analyzed.

5. Cost benefit analysis

“Cost Benefit Analysis (CBA) estimates and totals up the equivalent money value of the benefits and cost to the community of projects to establish they are worthwhile”.

2.2 SYSTEM STUDY

A detailed study to determine whether, to what extent, and how automatic data processing equipment should be used. We usually include an analysis of the existing system and the design of the new system, including the development of system specifications which provide a basis for the selection equipment.

Preliminary system study is the first stage of system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what actually the physical system is. In practice, the initial system study involves the preparation of a system proposal which lists the Problem Definition, Objectives of the Study, and Terms of reference for Study, Constraints, Expected benefits of the new system etc. in the light of the user requirements. The system proposal is prepared by the System Analyst (who studies the system) and places it before the user management. The management may accept the proposal and the cycle proceeds to the next stage. The management may also reject the proposal or request some modifications in the proposal.

In summary, we would say that system study phase passes through the following steps:

- Problem identification and project initiation
- Background analysis
- Inference or findings

2.3 EXISTING SYSTEM

In the existing system, the buyers go directly to the store and buy milk products directly from the nearby milk producers. It's generally difficult. They lose time, lose money, and so there are some difficulties. What should be available at a low price will increase in price when it reaches the store. Similarly, the milk farmer cannot sell his products directly. All transactions are handled directly. Currently everything is done on paper. One day owner and many users will have to use mobile call for product booking. The owner has to deal with a lot of things, and when that happens, it becomes very difficult for the owner. Everything interacts with the direct delivery boy to assign the order and give the money to the delivery boy. The delivery boy should meet the admin to know the order. Now the system using lots of manpower .

Limitations of existing system

- Time consuming
- Paper works
- Hard to find profit and loss
- Highly Expensive

To avoid all these limitations and make the system working more accurately it needs to be computerized.

2.4 PROPOSED SYSTEM

This project of ours is a very useful for all users, farmers, shops and milk company employees. The Admin, shop, and employee are website. User, delivery boy, and farmer modules are android application part. With the implementation of this project, users will be able to purchase all the products available with the admin at a reasonable price. Then all transactions can be done online. User can directly chat with farmers and buy farmer's products directly. The admin who is the owner of this project can give the salary to the employees and know the difficulties. Admin can know profit and loss of that company and see daily record. This application can help the small enterprises also. The user interface must be simple and easy to understand. This application will help in storing the products records like the dairy product, employee records, and, purchase information in an easy and well organized manner. This will be one of the projects that will help the owners of the dairy shop to maintain the inventory and the billing process with great ease. The aim of the project is to computerize the operations of collecting the sales details from the dealer and to book order for their next dispatching. The system is an automated system which handles all the process of the dairy product shop. This system facilitates the manufacturing company to organize its production schedules depending on the orders that are received from its recognized dealers.

2.5 FEASIBILITY STUDY

A feasibility study is a preliminary study undertaken to determine and document a project's viability. The results of this study are used to make a decision whether to proceed with the project. It is an analysis of possible alternative solutions to a problem and recommendation on the best alternative. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like.

The feasibility study environment enables all alternatives to be discussed and evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible. Normally, feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative that quantitative.

A feasibility study could be used to test a new working system, which could be used because:

- The current system may no longer suit its purpose
- Technological advancement may have rendered the current system obsolete
- The business is expanding allowing it to cope with extra workload
- Customers are complaining about the speed and quality of work the business provides.

Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Behavioral Feasibility

Technical Feasibility

Technical Feasibility study is performed to check whether the proposed system is technically feasible or not. Technical feasibility centers on the existing computer system (hardware, software, etc.) and to what extent it can support the proposed addition. This involves financial consideration to accommodate technical enhancement. This system is technically feasible. All the data are stored in files. The input can be done through dialog boxes which are both interactive and user friendly. Hard copies can be obtained for future use, by diverting the documents to a printer. Windows serves as the platform for the new system.

Operational Feasibility

Operational Feasibility study is performed to check whether the system is operationally feasible or not. Using command buttons throughout the application programs enhances operational feasibility. So, maintenance and modification is found to be easier. The purpose of this feasibility is to determine whether the new system will be used if it is developed and implemented. Usually, people are reluctant to changes that come in their progression. Hence an additional effort is to be made to train and educate the users on the new way of the system.

In this project the system is well coded and designed such that when it is compiled it is well developed and installed. We can do maintenance and modifications easily. There is no difficulty in implementing the system and the proposed system is user friendly and functionally reliable and thus it is operational feasible.

Economic Feasibility

Economic Feasibility Study is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. This analysis phase determines how much cost is needed to produce the proposed system.

This system is economically feasible since it does not require any initial setup. It requires only network connection and the cost needed to host the site.

Behavioral Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. An estimate should be made about the reaction of the user staff towards the development of the computerized system. The introduction of a candidate system requires special effort to educate, sell and train the staff for conducting the business. This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?
- This system is behavior feasible since the system sufficiently support the users and the system does not cause any harm to the users.

The project would be beneficial because it satisfies the objectives when developed and installed. All the behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

2.6 SYSTEM ENVIRONMENT

SOFTWARE SPECIFICATION:

One of the most difficult tasks is selecting software for the system. Once the system requirement is found out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

Operating System : Windows 8 or above

Front End : python/ Android

Back End : MySQL

Programming Language

Language : Python, Android

Browser : Internet Explorer/ Mozilla Firefox/Google Chrome

HARDWARE SPECIFICATION:

Selection of hardware is very important for the working of any software. When selecting the size and capacity requirements are also important.

Processor : i3 or above

Hard disk : 500 GB and above

RAM : 4 GB and above

Monitor : 14" LCD or Above

Keyboard : 104 keys

Mouse : Optical/scroll mouse

Mobile: Android supported mobile phone

2.7 FRONT END

ANDROID

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touch screen mobile devices such as smart phones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, though its most widely used version is primarily developed by Google. It was unveiled in November 2007, with the first commercial Android device, the HTC Dream, being launched in September 2008.

At its core, the operating system is known as Android Open-Source Project (AOSP) and is free and open-source software (FOSS) primarily licensed under the Apache License. However, most devices run on the proprietary Android version developed by Google, which ship with additional proprietary closed-source software pre-installed, most notably Google Mobile Services (GMS) which includes core apps such as Google Chrome, the digital distribution platform Google Play, and the associated Google Play Services development platform. Firebase Cloud Messaging is used for push notifications. While AOSP is free, the "Android" name and logo are trademarks of Google, which imposes standards to restrict the use of Android branding by "uncertified" devices outside their ecosystem.

Over 70 percent of smart phones based on Android Open-Source Project run Google's ecosystem (which is known simply as Android), some with vendor-customized user interfaces and software suites, such as Touch Wiz and later One UI by Samsung and HTC Sense. Competing ecosystems and forks of AOSP include Fire OS (developed by Amazon), ColorOS by OPPO, OriginOS by Vivo, MagicUI by Honor, or custom ROMs such as Lineage OS.

The source code has been used to develop variants of Android on a range of other electronics, such as game consoles, digital cameras, portable media players, and PCs, each with a specialized user interface. Some well-known derivatives include Android TV for televisions and Wear OS for wearable's, both developed by

Google. Software packages on Android, which use the APK format, are generally distributed through proprietary application stores like Google Play Store, Amazon Appstore (including for Windows 11), Samsung Galaxy Store, Huawei AppGallery, Cafe Bazaar, GetJar and Aptoide, or open-source platforms like F-Droid.

Android has been the best-selling OS worldwide on smart phones since 2011 and on tablets since 2013. As of May 2021, it had over three billion monthly active users, the largest installed base of any operating system in the world, and as of January 2021, the Google Play Store featured over 3 million apps. Android 13, released on August 15, 2022, is the latest version, and the recently released Android 12.1/12L includes improvements specific to foldable phones, tablets, desktop-sized screen and Chrome books.

Features

Interface

Android's default user interface is mainly 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, along with a virtual keyboard. Game controllers and full-size physical keyboards are supported via Bluetooth or USB. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware, such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

Home screen

Android devices boot to the home screen, the primary navigation and information "hub" on Android devices, analogous to the desktop found on personal

computers. Android home screens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as a weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on Google Play and other app stores can extensively re-theme the home screen, and even mimic the look of other operating systems, such as Windows Phone. Most manufacturers customize the look and features of their Android devices to differentiate themselves from their competitors.

Status bar

Along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be pulled (swiped) down from to reveal a notification screen where apps display important information or updates, as well as quick access to system controls and toggles such as display brightness, connectivity settings (WiFi, Bluetooth, cellular data), audio mode, and flashlight. Vendors may implement extended settings such as the ability to adjust the flashlight brightness.

Notifications

Notifications are "short, timely, and relevant information about your app when it's not in use", and when tapped, users are directed to a screen inside the app relating to the notification. Beginning with Android 4.1 "Jelly Bean", "expandable notifications" allow the user to tap an icon on the notification in order for it to expand and display more information and possible app actions right from the notification.

App lists

An "All Apps" screen lists all installed applications, with the ability for users to drag an app from the list onto the home screen. The app list may be accessed using a gesture or a button, depending on the Android version. A "Recents" screen, also known as "Overview", lets users switch between recently used apps.

The recent list may appear side-by-side or overlapping, depending on the Android version and manufacturer.

Navigation buttons

Many early Android OS smart phones were equipped with a dedicated search button for quick access to a web search engine and individual apps' internal search feature. More recent devices typically allow the former through a long press or swipe away from the home button.

The dedicated option key, also known as menu key, and its on-screen simulation, is no longer supported since Android version 10. Google recommends mobile application developers to locate menus within the user interface. On more recent phones, its place is occupied by a task key used to access the list of recently used apps when actuated. Depending on device, its long press may simulate a menu button press or engage split screen view, the latter of which is the default behavior since stock Android version 7.

Split-screen view

Native support for split screen view has been added in stock Android version 7.0 Nougat.

The earliest vendor-customized Android-based smartphones known to have featured a split-screen view mode are the 2012 Samsung Galaxy S3 and Note 2, the former of which received this feature with the premium suite upgrade delivered in TouchWiz with Android 4.1 Jelly Bean.

Charging while powered off

When connecting or disconnecting charging power and when shortly actuating the power button or home button, all while the device is powered off, a visual battery meter whose appearance varies among vendors appears on the screen, allowing the user to quickly assess the charge status of a powered-off without having to boot it up first. Some display the battery percentage.

Applications

Many, to almost all, Android devices come with preinstalled Google apps including Gmail, Google Maps, Google Chrome, YouTube, Google Play Music, Google Play Movies & TV, and many more.

Applications ("apps"), which extend the functionality of devices (and must be 64-bit, are written using the Android software development kit (SDK) and,

often, Kotlin programming language, which replaced Java as Google's preferred language for Android app development in May 2019, and was originally announced in May 2017. Java is still supported (originally the only option for user-space programs, and is often mixed with Kotlin), as is C++. Java or other JVM languages, such as Kotlin, may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API).

The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Other development tools are available, including a native development kit (NDK) for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled a framework based on Apache Cordova for porting Chrome HTML 5 web applications to Android, wrapped in a native application shell. Additionally, Firebase was acquired by Google in 2014 that provides helpful tools for app and web developers.

PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Features

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural

programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type `print("Hello World")`. It will take only one line to execute, while Java or C takes multiple lines.

Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables

programmers to develop the software for several competing platforms by writing a program only once.

Free and Open Source

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

Large Standard Library

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQt5, Tkinter, Kivy are the libraries which are used for developing the web application.

HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the meaning and structure of web content. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for its appearance.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as and <input> directly introduce content into the page. Other tags such as <p> and </p> surround and provide information about document text and may include sub-element tags. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former

maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[2] A form of HTML, known as HTML5, is used to display video and audio, primarily using the <canvas> element, together with JavaScript.

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development.[13] Java is still supported, as is C++.

PYCHARM

PyCharm is an integrated development environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm is developed by the Czech company JetBrains.

It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. PyCharm Community Edition is less extensive than the Professional Edition.

2.8 BACK END

MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).[8] In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

Features

MySQL is a relational database management system (RDBMS) based on the SQL (Structured Query Language) queries. It is one of the most popular languages for accessing and managing the records in the table. MySQL is open-source and free software under the GNU license. Oracle Company supports it.

The following are the most important features of MySQL:

Relational Database Management System (RDBMS)

MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.

Easy to use

MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

It is secure

MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

Client/ Server Architecture

MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

SYSTEM DESIGN

3.1 SYSTEM DESIGN

System Architecture

Design is a multi-step process that focuses on data structure, software architecture, procedural details and interface between the modules. The design process also translates the requirements into the representation of the software that can access for quality before coding begins. Design is the only way that can accurately translate a customer's requirements into a finished software product.

The most creative and challenging phase of the system development is the system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development. System design goes through two phases of development:

- Logical design
- Physical design

Logical Design

The part of the design process that is independent of any specific hardware or software platform is referred to as logical design. During logical design, all functional features of the system chosen for development in analysis phase are described independently of any computer platform. Logical design concentrates on the business aspects of the system and tends to be oriented to a high level of specificity. During logical design of the proposed system, all the functional features of the system which are identified during the analysis phase is collected and properly specified independent of any hardware and software specifications.

Physical Design

During physical design the logical specifications of the system from logical design are transferred into technology specific details from which all programming and system construction can be accomplished.

It produces the working system by defining the design specification that specify exactly what the candidate system does. It is concerned with the user interface design, process design and data design.

It consists of the following steps:

- Specifying the input/ output media, designing the database, and specifying backup procedures.
- Planning system implementation.
- Devising a test and implementation plan, and specifying any new hardware and software.
- Updating costs, benefits, conversion dates, and system constraints.

3.2 INPUT DESIGN

In input design, user originated inputs are converted into computer-based format. In output design, the emphasis is on producing the hard copy of the information requested or displaying the output on the screen in a predefined format. The following features have been incorporated into the input design of the proposed system.

Easy Data Input

Appropriate messages are provided in the message area, which prompts the user entering the right data. Erroneous data inputs are checked at the end of each screen entry.

Data Validation

The input data is validated to minimize errors in data entry. For certain, data specific codes have been given and validation is done which enables the user to enter the required data or correct them if they entered wrong inputs.

User Friendliness

User is never left in a state of confusion as to what is happening, instead appropriate error and acknowledge messages are sent. Flutter alert messages are being used to indicate the error, warning messages.

Consistent Format

A fixed format is adopted for displaying the alert messages. Every screen has line, which displays the operation that can be performed after the data entry. They are normally done at the touch of a key.

Interactive Dialogue

The system engages the user in an interactive dialogue. The system is able to extract missing or omitted information from the user by directing the user through appropriate messages, which are displayed.

3.3 OUTPUT DESIGN

Output design generally refers to the results and information generated by the system. For many end users, output is the main reason for developing the system and the basis on which they evaluate the usefulness of application.

The objective of a system finds its shape in terms of the output. The analysis of the objective of a system leads to determination of outputs. Output of a system can take various forms. The most common are reports, screen displays, printed form, graphical drawings etc. The output also varies in terms of their contents, frequency, timing and format. The output from a system is the justification for its existence. If the outputs are inadequate in anyway, the system itself is inadequate. The basic requirement of output is that it should be accurate, timely and appropriate in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output, so that the objective of the system is met in the best possible manner.

When designing output, the system analyst must accomplish things like, to determine what the information to present, to decide whether to display or print the information and select the output medium to distribute the output to intended recipients.

External outputs are those, whose destination will be outside the organization and which requires special attention as a project image of the organization. The output is those whose destination within the organization. It is to be carefully designed, as they are the user's main interface with the system. Interactive output are those which the user uses to communicate directly with the computer.

3.4 DATA FLOW DIAGRAM

The database may be defined as an organized collection of related information. The organized information serves as a base from which further recognizing can be retrieved desired information or processing the data. The most important aspect of building an application system is the design of tables.

The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines.

- Bubbles: Represent the data transformations.
- Lines: Represents the logic flow of data.

Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This data flow analysis tells a great deal about organization objectives are accomplished.

Data flow analysis is studying the use of data in each activity. It documents this finding in DFD's. Dataflow analysis give the activities of a system from the viewpoint of data where it originates show they are used or hanged or where they go, including the stops along the way from their destination. The components of data flow strategy span both requirements determination and system's design. The first part is called data flow analysis.

As the name suggests, we didn't use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

Constructing a DFD

Several rules of thumb are used in drawing DFDs: -

- Process should be named and numbered for easy reference.
- The direction of flow is from source to destination, although they may flow back to a source. One way to indicate this is to draw a long flow line back to the source. An alternative way is to repeat the source symbol as a destination.
- When a process is exploded into lower-level details, they are numbered.
- The names of data stores, sources, and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalize.

A level 0 DFD ,also called a context level, represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level1 DFD. Any process, which is complex in Level1, will be further represented into sub functions in the next level.i.e, Level 2 DFD is a means of representing a system at any level of detail with a graphic network of symbols showing data flows,data stores, data process, sources or destination

The DFD is designed to aid communication. DFD shows the minimum contents of data stores. In order to show what happens within the given process, then the detailed explosion of that process is shown. The DFD methodology is quite effective, especially when the required design is unclear and the user and the analyst need a notational language for communication.

Context Diagram

The top-level diagram is often called a “context diagram”. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries.

Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

Data Flow Diagrams are made up of a number of symbols which represents system components.

Data flow modeling uses four kinds of symbols.

Process

Process shows work of the system. Each process has one or more data inputs and produce one or more data outputs. Processes are represented by circles in DFDs.

Data Stores

Data stores are the repositories of data. Process can enter data into store or retrieve the data from data store. Data stores are represented by parallel lines which may be depicted horizontally or vertically.

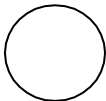

Data flow

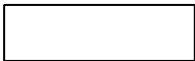
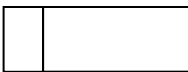
The arrows represent data flow. A data flow is data in motion. Data flow represents an input of data to a process or the output of the data from a process.

External Entities

External entities are outside the system but they either supply input to the system or use other system outputs. They are entities on which the designer has control. External entities that supply data into the system are sometimes called source. External entities that use the system are called sinks, represented by rectangles.

Data Flow Diagram Symbols

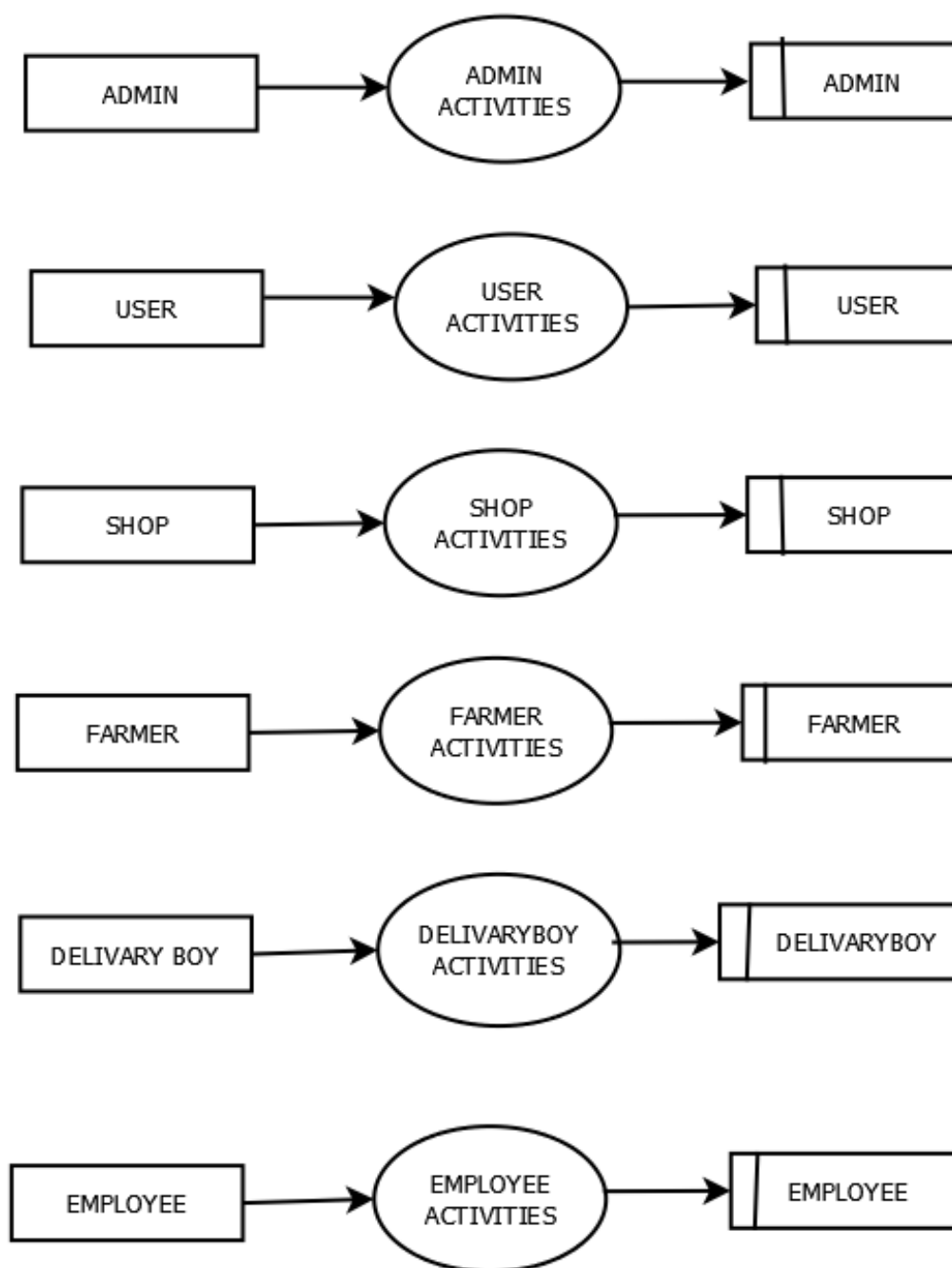
Symbol	Name	Function
	Process	Performs some transformation of input data to yield output data.
	Data Flow	Used to connect processes to each other, to sources or Sinks, the arrow head indicates direction of data flow.

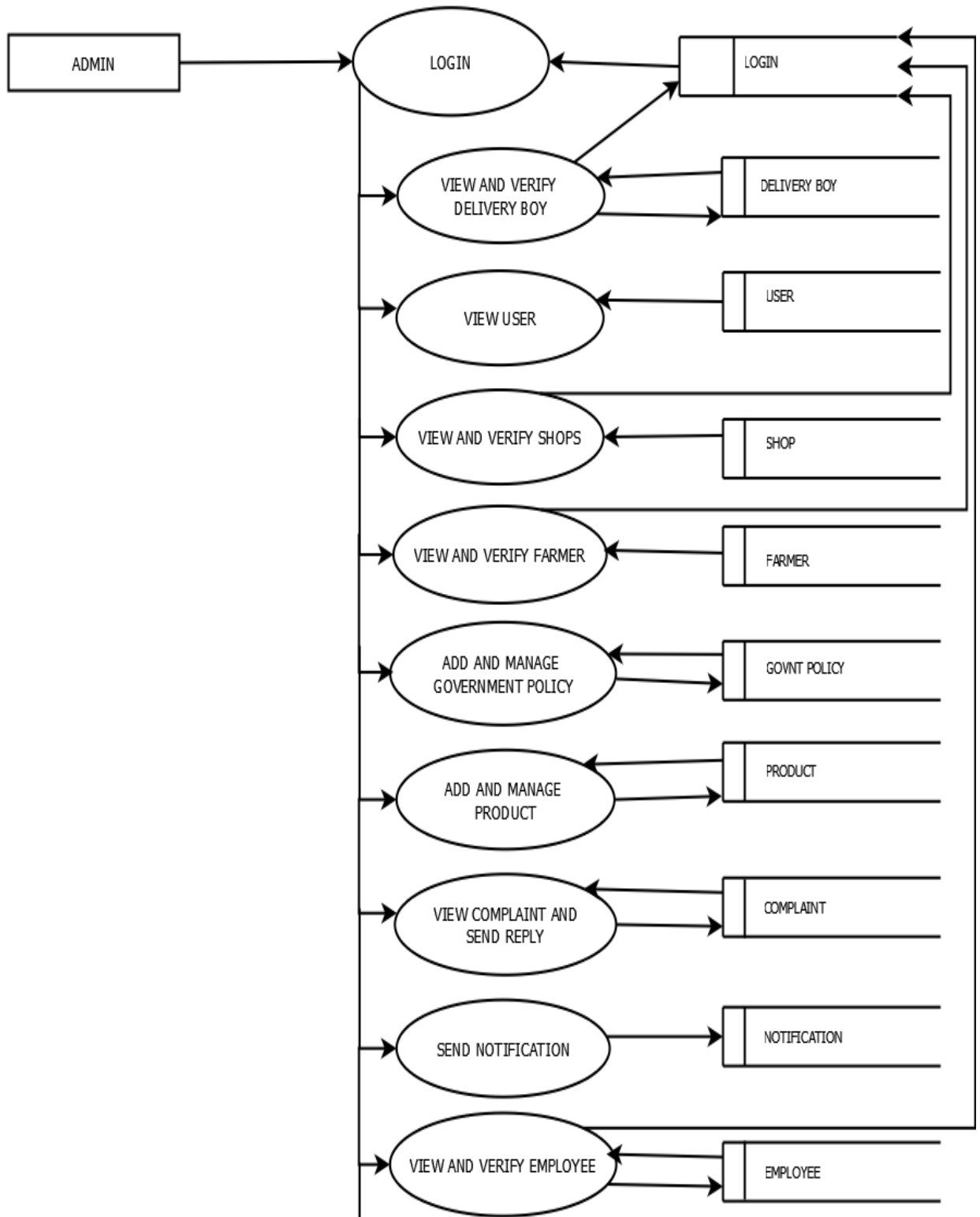
	Source or sink (External Entity)	A source of system inputs or sinks of system outputs.
	Data Store (Internal Storage)	A repository of data.

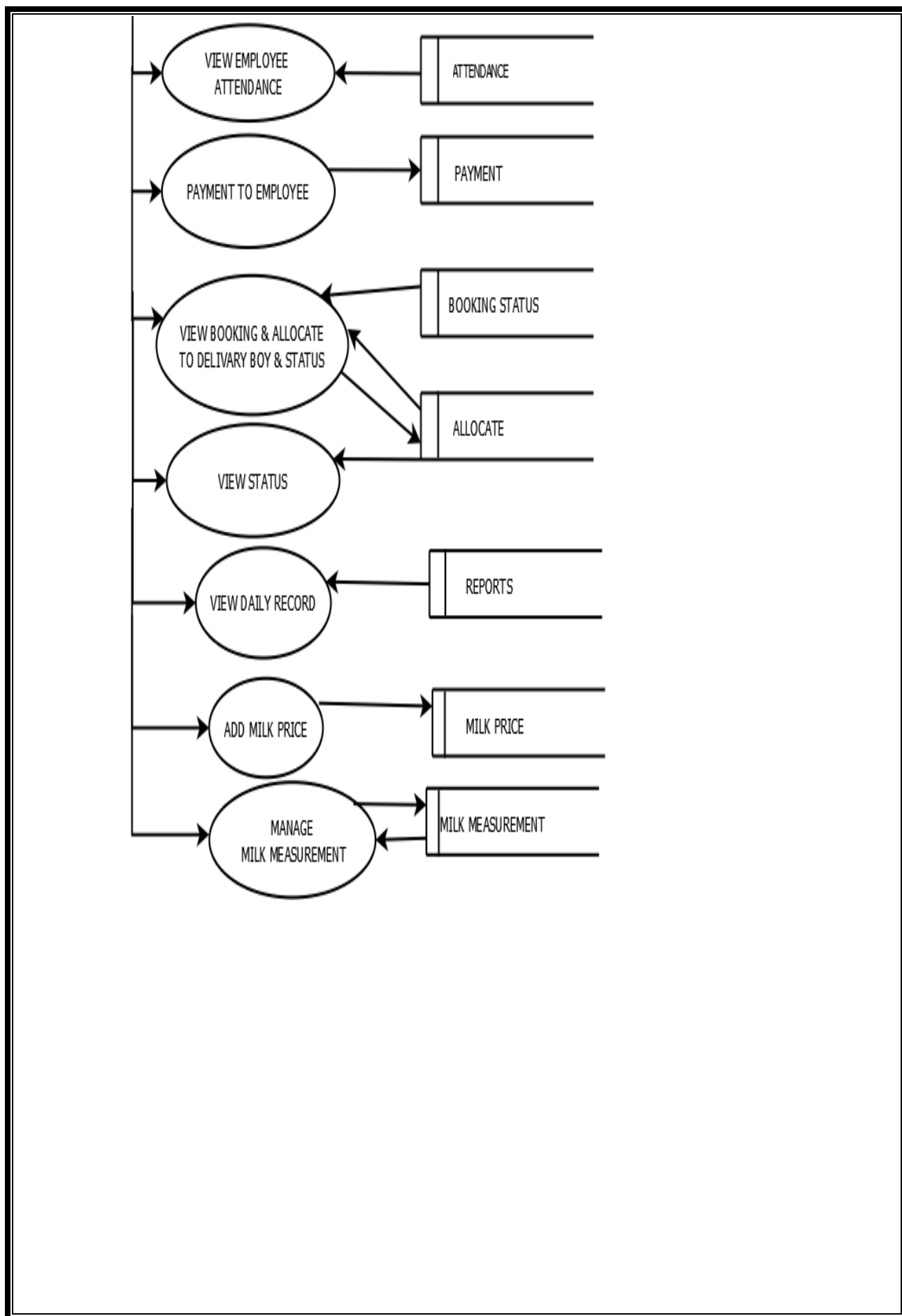
DATA FLOW DIAGRAM

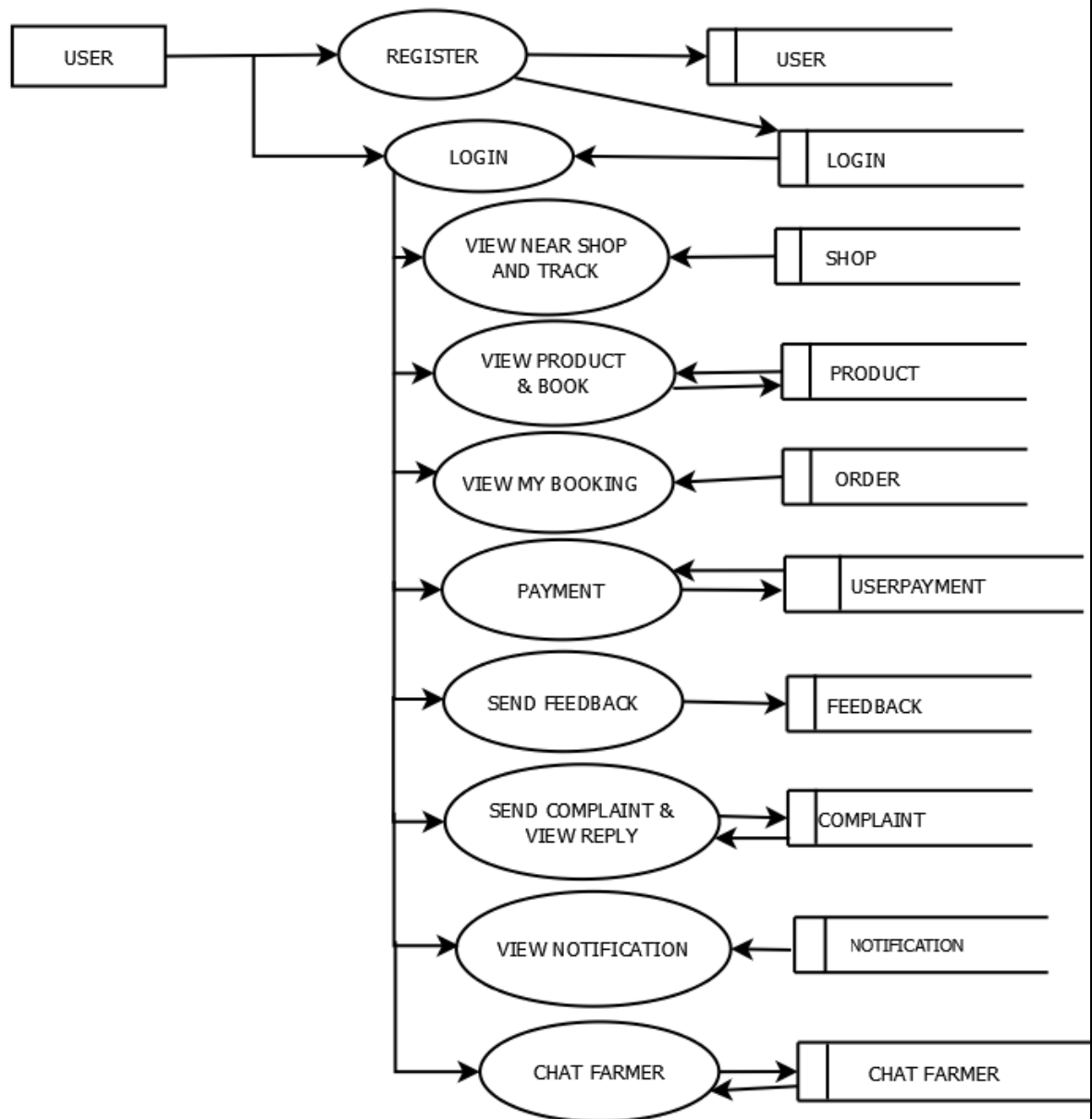
LEVEL 0

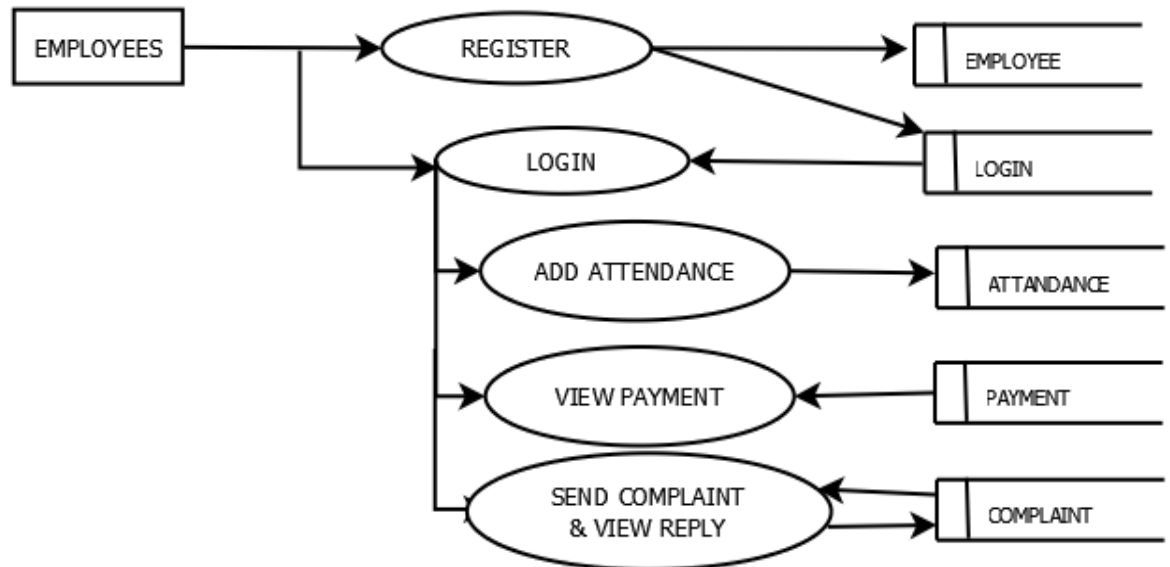
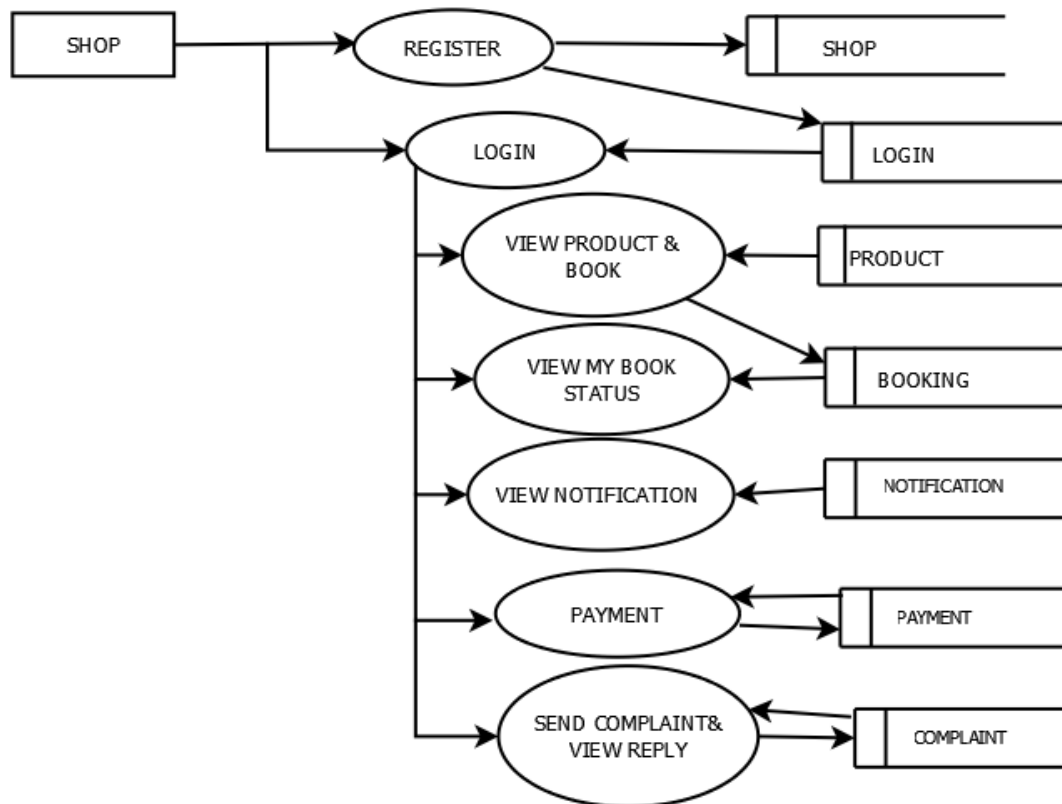


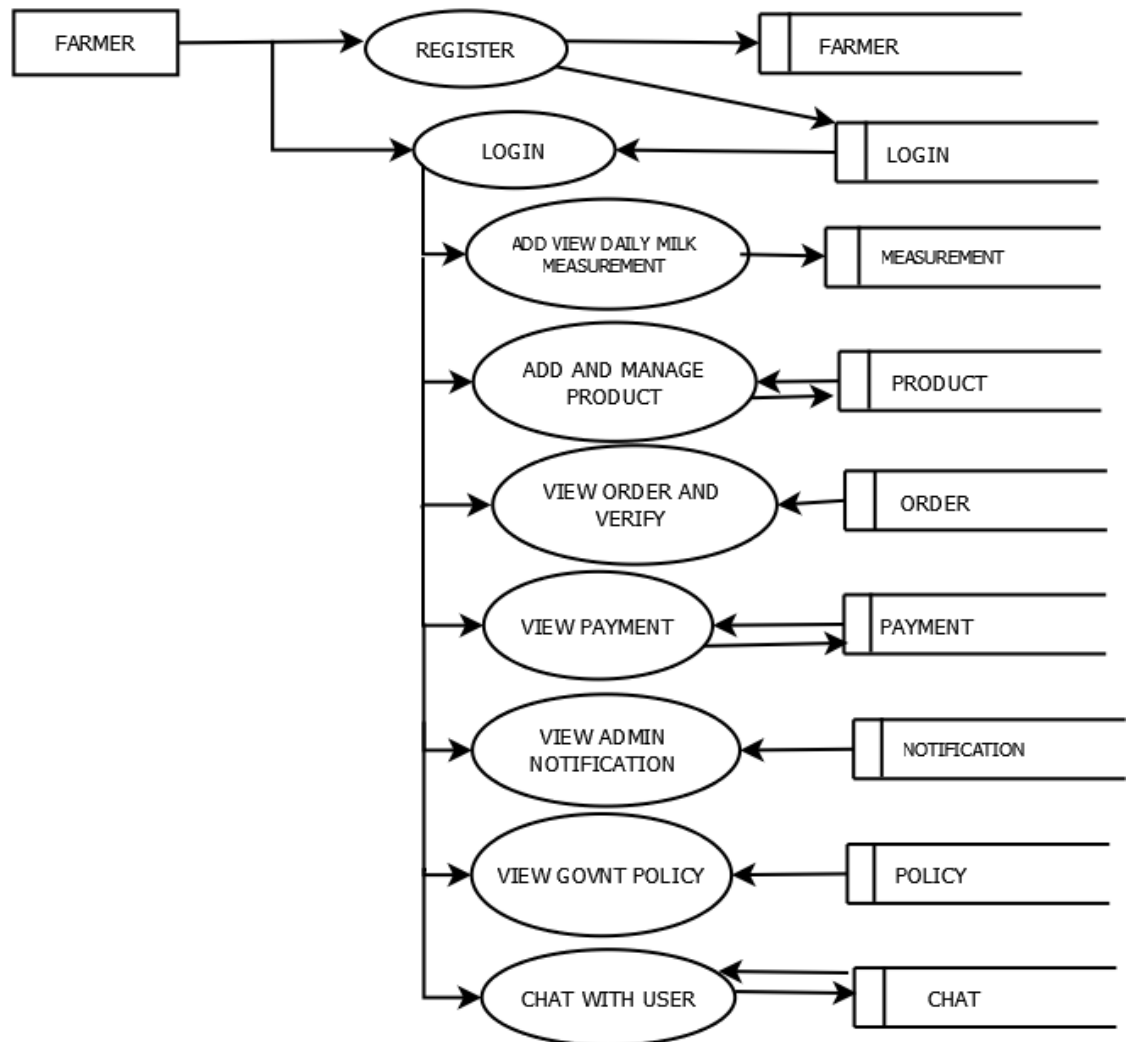
LEVEL 1

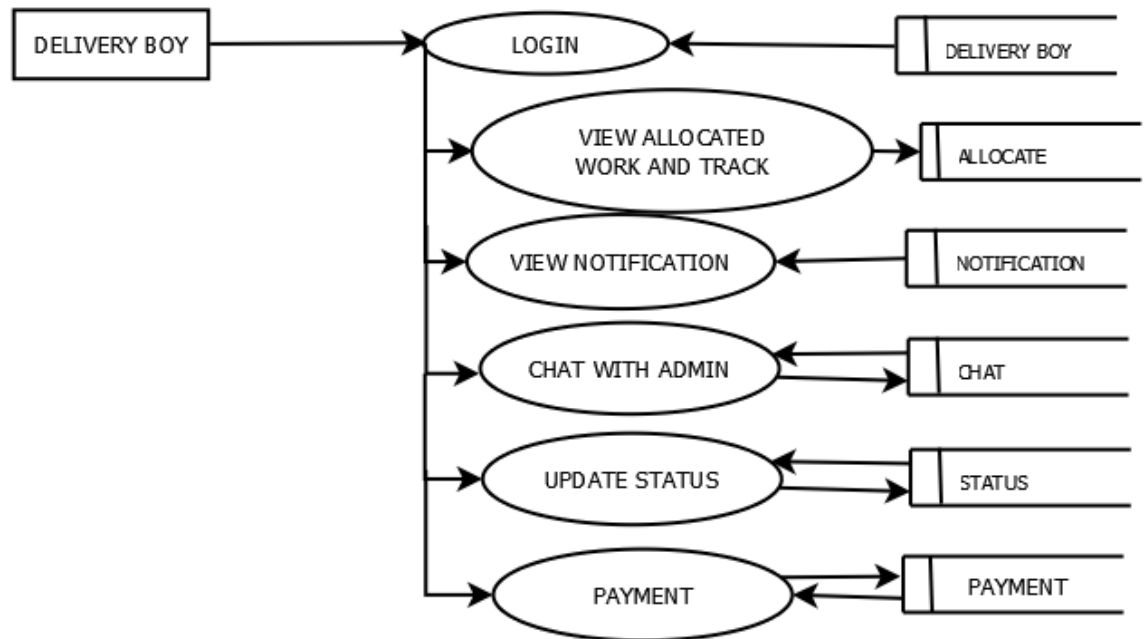
LEVEL1.1: ADMIN



LEVEL 1.2: USER

LEVEL 1.3:EMPLOYEE**LEVEL 1.4: SHOP**

LEVEL 1.5:FARMER

LEVEL 1.6: DELIVERY BOY

3.5 DATABASE DESIGN

The general objective is to make information necessary, quick, inexpensive and flexible for the user. Database allows the data to be protected and organized separately from other resources. A database is a collection of interested data stored with minimum redundancy to serve many users quickly and efficiently. The general objectives considered in database design are controlled redundancy, ease of learning and use, data independency, more information at low cost, accuracy and integrity, recover from failure, privacy and security performance.

In a database environment, the Database Management System (DBMS) is the software that provides the interface between the data file on a disk and the management, they differ in the way they structure data. The three types of data structures are hierarchical, network and relational. Here we use relational structuring in which all data and relationships are presented in a flat, two- dimensional table called a relation. A relation is equivalent to file.

Data structuring is refined through a process called normalization. Data are grouped into simplest way possible, so that later changes can be made with a minimum impact on data structures. Based on the requirements determined during the definition phase of project life cycle, the data elements describing the entity were determined. They are later submitted to normalization to remove redundancy and to optimize them.

Database objectives:

- Eliminate redundant data spontaneously
- Integrate existing datafiles
- Share data among all users
- Incorporate changes identifies easily and quickly
- Simplifies the use of datafiles
- Lower the cost of storing and retrieving data
- Improve accuracy and consistency
- Provide data security from unauthorized use
- Exercise central control over standards

NORMALIZATION

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multistep process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes:

- Eliminating redundant (useless) data
- Ensuring data dependencies make sense i.e., data is logically stored.

Problem without Normalization

Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updating and Deletion Anomalies are very frequent if database is not normalized. To understand these anomalies let us take an example of Student table.

- **Updation Anomaly:** An update anomaly exists when one or more instances of duplicated data is updated but not all and thus the data will become inconsistent.
- **Insertion Anomaly:** An Insertion anomaly occurs when certain attributes cannot be inserted into the database without the presence of other attributes.
- **Deletion Anomaly:** A record of data can legitimately be deleted from a database, and the deletion can result in the deletion of the only instance of other, required data.

In our design we have the database normalized up to third normal form.

- First normal form
- Second normal form
- Third normal form

First Normal Form (1NF)

As per First Normal Form, no two Rows of data must contain repeating group of information i.e., each set of columns must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

Second Normal Form (2NF)

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.

Third Normal Form (3NF)

Third Normal form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this transitive functional dependency should be removed from the table and also the table must be in Second Normal form. In this project the tables are in Third normal form and thus there does not exist any transitive dependency

3.6 TABLE DESIGN

Table1: LOGIN

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Primary key	To store login id
username	varchar(50)	Not Null	To store Username
password	varchar(50)	Not Null	To store Password
type	varchar (25)	Not Null	To store Type

Table2: USER

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Primary key	To store login id
uid	int (11)	Not Null	To store user id
name	varchar (11)	Not Null	To store user name
age	varchar (25)	Not Null	To store age
gender	varchar(50)	Not Null	To store gender
phone	Bigint (10)	Not Null	To store number
place	Varchar (50)	Not Null	To store place
email	Varchar (50)	Not Null	To store email

Table3: COMPLAINT

FIELD	TYPE	CONSTRAINT	DESCRIPTION
complaint_id	int (11)	Primary key	To store complaint id
lid	int (11)	Default Null	To store login id
complaint	varchar (25)	Default Null	To store complaint id
date	varchar (25)	Default Null	To store date
reply	varchar (25)	Default Null	To store reply

Table4: DELIVERY BOY

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Primary key	To store login id
did	int (11)	Not Null	To store delivery boyid
name	varchar (25)	Not Null	To store name
age	varchar (25)	Not Null	To store age
license	varchar(50)	Not Null	To store license file
contact	bigint (10)	Not Null	To store contact number
mail	varchar (50)	Not Null	To store mail address
address	varchar (50)	Not Null	To store address

Table5: ASSIGN

FIELD	TYPE	CONSTRAINT	DESCRIPTION
assign_id	int (11)	Primary key	To store assign id
order_id	int (11)	Not Null	To store order id
delboy_id	int (11)	Not Null	To store delivery boy id
date	varchar (25)	Not Null	To store date
status	varchar (25)	Not Null	To store to status

Table6: ATTANDANCE

FIELD	TYPE	CONSTRAINT	DESCRIPTION
Attendance_id	int (11)	Primary key	To store attendanceid
emp_id	int (11)	Not Null	To store employeeid
date	date	Not Null	To store date
Attendance	Varchar (50)	Not Null	To store attendance

Table7: CHAT

FIELD	TYPE	CONSTRAINT	DESCRIPTION
chat_id	int (11)	Primary key	To store chat id
from_id	int (11)	Not Null	To store sender id
date	date	Not Null	To store date
to_id	int(11)	Not Null	To store receiver id
status	varchar (25)	Not Null	To store status
messsage	varchar (25)	Not Null	To store messages

Table8: FARMER

FIELD	TYPE	CONSTRAINT	DESCRIPTION
farmer_id	int (11)	Primary key	To store farmer id
lid	int (11)	Default Null	To store login id
farmerame	varchar (50)	Default Null	To store farmer name
farmerage	int (10)	Default Null	To store farmer age
gender	char (10)	Default Null	To store farmer gender
place	varchar (50)	Default Null	To store place
email	varchar (50)	Default Null	To store email
phone	bigint (11)	Default Null	To store phone number

Table9: EMPLOYEE

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	foreign key	To store login id
empid	int (11)	Primary key,NotNull Auto increment	To store login_id
empname	varchar (25)	Default Null	To store employee name
empage	varchar (25)	Default Null	To store age
gender	varchar (25)	Default Null	To store gender
contact	varchar (25)	Default Null	To store contact number
email	varchar (25)	Default Null	To store email
address	varchar (25)	Default Null	To store address
department	varchar (25)	Default Null	To store department of the employee
pin	int (11)	Default Null	To store pincode
post	varchar(50)	Default Null	To store post

Table10: FEEDBACK

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Foreign key	To store login id
feedbackid	int (11)	Primary key	To store feedback id
feedback	text	Default Null	To store feedback
date	date	Default Null	To store date

Table 11: LOCATION

FIELD	TYPE	CONSTRAINT	DESCRIPTION
locationid	int (11)	Primary key	To store location id
uid	int (11)	Foreign key	To store user id
latitude	varchar (25)	Default Null	To store latitude
longitude	varchar (25)	Default Null	To store longitude

Table 12: MILK MEASUREMENT

FIELD	TYPE	CONSTRAINT	DESCRIPTION
measurement_id	int (11)	Primary key	To store measurement id
quantity	varchar (25)	Default Null	To store quantity of milk
price	float	Default Null	To store milk price
date	varchar(50)	Default Null	To store date
farmerid	int(11)	Foreign key	To store farmer id

Table 13: MILK PRICE

FIELD	TYPE	CONSTRAINT	DESCRIPTION
id	int (11)	Primary key	To store milk price id
Price	float	Default Null	To store price
date	date	Default Null	To store date

Table 14: NOTIFICATION

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Foreign key	To store login id
notificationid	int (11)	Primary key,Not Null, Autoincrement	To store notification id
notifications	longtext	Default Null	To store notifications
date	varchar (25)	Default Null	To store date
satatus	varchar (25)	Default Null	To store notification status

Table 15: ORDER

FIELD	TYPE	CONSTRAINT	DESCRIPTION
order_id	int (11)	Primary key, Not Null Auto increment	To store order id
lid	int (11)	Default Null	To store login id
date	varchar (30)	Default Null	To store dat
total	double (25)	Default Null	To store total
status	varchar (30)	Default Null	To store status

Table 16: ORDER ITEM

FIELD	TYPE	CONSTRAINT	DESCRIPTION
item_id	int (11)	Primary key, Not Null, Auto increment	To store item id
order_id	int(11)	Default Null	To store order id
product_id	int (11)	Default Null	To store product id
date	date	Default Null	To store date
quantity	int(11)	Default Null	To store quantity
status	varchar (50)	Default Null	To store status

Table 17: PAYMENT

FIELD	TYPE	CONSTRAINT	DESCRIPTION
lid	int (11)	Foreign key	To store login id
details	varchar (50)	Default Null	To store details
amount	float	Default Null	To store amount
type	varchar (25)	Default Null	To store type
date	date	Default Null	To store date

Table18: POLICY

FIELD	TYPE	CONSTRAINT	DESCRIPTION
policyid	int (11)	Primary key,Not Null, Auto increment	To store policy id
policyname	varchar (25)	Default Null	To store policy name
details	text	Default Null	To store policy details
for_more	text	Default Null	To store more detail

Table 19: PRODUCT

FIELD	TYPE	CONSTRAINT	DESCRIPTION
productid	int (11)	Primary key	To store product id
lid	int(11)	Foreign key	To store login id
product	varchar(50)	Default Null	To store product name
price	float	Default Null	To store productprice
image	varchar (1000)	Default Null	To store product image
stock	varchar (500)	Default Null	To store product stock
details	text	Default Null	To store productdetails

Table 20: SHOP

FIELD	TYPE	CONSTRAINTS	DESCRIPTION
shop_id	int	Primary key ,Not Null,Auto Increment	To store shop id
lid	int	Foreign key	To store login id
latitude	varchar(60)	Default Null	To store latitude
longitude	varchar(60)	Default Null	To store longitude
name	varchar	Default Null	To store shop name
Pin	int	Default Null	To store shop pin code
place	varchar	Default Null	To store shop place
phone	bigint	Default Null	To store shop id
email	varchar	Default Null	To store email
post	varchar	Default Null	To store post

Table 21: USER PAYMENT

FIELD	TYPE	CONSTRAINTS	DESCRIPTION
userid	int(11)	Primary key ,Not Null, Auto Increment	To store user id
oid	int(11)	Foreign key	To store order id
amount	double	Default Null	To store amount
date	date	Default Null	To store date

SYSTEM DEVELOPMENT

MODULE DESCRIPTION

The platform has 4 major modules and each module has its features to carry out.

MODULES

1. Admin
2. Employee
3. User
4. Shop
5. Delivery boy
6. Farmer

1.Admin functions

Admin is a main part of the system . Admin handle other modules.

- Login
- View and verify delivery boy
- View user
- Add and manage government policies
- View complaint and send reply
- View and verify farmer
- View and verify shop
- Send notification
- Add and manage products
- View booking and allocate to delivery boy
- Add milk price and measurement
- View daily records
- View and verify Employee
- View Employee attendance
- Payment to Employee
- Chat with delivery boy

2. Employees functions

The employee is one of the module in this system. The admin give work to employees and they do the work and add their attendance .with the base of attendances admin will give salary to the employees.

- Login
- Register
- Add Attendance
- View payment
- Complaint& reply

3. User functions

User is the android application part of the system. The users are purchase products from farmers and admin

- Login
- Register
- View shops & track
- View products and book
- View shop bookings
- View my booking status
- Payment
- Send feedback
- Send complaint and view reply
- View notification
- Chat with farmer

4.Shop functions

Shop is a module purchase bulk load of products from the admin.

- Login
- Register
- View product and book
- View notification
- View my booking status
- Payment
- Send complaint & view reply

5.Delivery boy functions

Delivery boy collect order from the admin and deliver order to the shops and users.

- Login
- Register
- View allocated work and track
- View notification
- Update status
- Chat with admin

6.Farmer functions

Farmer gives milk to admin. Produce their own products and sell the product to the users through the system

- Register
- Login
- Add daily milk measurement
- Add and manage product
- View order & verify
- View government policy
- Chat with user

SYSTEM TESTING

5.1 SYSTEM TESTING

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. However not restricted to being performed after the development phase is complete, but this is to carry out in parallel with all stages of system development, starting with requirements specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A project is said to be incomplete without proper testing.

The testing processors focus on the logical internal of the software and ensuring that all statements have been tested and on the functional externals, i.e, conducting tests to uncover errors and ensure that defined input will produce actual results.

System testing is a process of checking whether the developed system is working according to the original objectives and requirements. The proposed system has been tested experimentally with test data so as to ensure that system works according to the required specification. The system is found working, and is tested with actual data and checked the performance. The performance of the system was good and satisfactory. The testing objectives are summarized in the following steps:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one has high probability of finding an as yet undiscovered error.

TESTING PRINCIPLES:

All tests should be traceable to customer requirements. The focus of testing will shift progressively from programs. Exhaustive testing is not possible. To be more effective, testing should be one, which has probability of finding errors. The following are attributes of good test

- A good test has a high probability of finding error
- A good test is not redundant.

5.2 LEVELS OF TESTING

Testing is the process of finding bugs in a program. It helps to improve the quality of the software. It has to be done thoroughly and with the help of specialist testers. System testing is a process of checking whether the developed system is working according to the objectives and requirements.

In construction management application first, the whole application divided into small units then tested each units individually after that combines all the units and then whole system tested and at last the validation testing done for checking data accuracy, data completion, data transformation, data quality and ensures that this application met all the needs.

The testing procedure that has been used in the system is as follows:

- Unit Testing
- Integration Testing
- Validation Testing

UNIT TESTING

The first level of testing is called as unit testing. Here the different modules are tested and the specification produced during design for the modules. Unit testing is essential for verification of the goal and to test the internal logic of the modules. Unit testing is conducted to different modules of the project. Errors were noted down and corrected down immediately and the program clarity was increased. The testing was carried out during the programming stage itself. In this step each module is found to be working satisfactory as regard to be expected out from the module.

In construction management application unit testing has the goal of discovering errors in the individual four modules of the system where it is concerned with the decision logic, control flow, recovery procedures, throughput, capacity & timing characteristics of the entire system by testing individually.

INTEGRATION TESTING

The second level of testing includes integration testing. It is a systematic testing of constructing structure. At the same time tests are conducted to uncover errors with the interface. It need not to be the case, that software whose modules when run individually showing results will also show perfect results when run as a whole.

The individual modules are tested again and the results are verified. The goal is to see if the modules integrated between the modules. This testing activity can be considered as testing the design and emphasizes on testing modules interaction.

VALIDATION TESTING

The next level of testing is validation testing. Here the entire software is tested. The reference document for this process is the requirement and the goal are to see if the software meets its requirements.

The requirement document reflects and determines whether the software functions as the user expected. At culmination of integration testing, software is completely assembled as a package and corrected and a final series of software test validation test begins. The proposed system under construction has been tested by using validation testing and found to be working satisfactory.

Data validation checking is done to see whether the corresponding entries made in different tables are done correctly. Proper validation checks are done in case of insertion and updating of tables, in order to see that no duplication of data has occurred. If any such case arises proper warning message will be displayed. Double configuration is done before the administrator deletes a data in order to get positive results and to see that data have been deleted by accident.

Validation testing done in the construction management application and it is a dynamic testing, where ensured that “developed the product right” and checks that the application meets the needs mentioned in the requirement analysis phase.

SYSTEM IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION

System implementation is the final phase that is, putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. Implementation involves the conversion of a basic application to complete replacement with a computer system. It is the process of converting to a new or revised system design into an operational one. During the phase, the products structure, its undergoing data structures, the general algorithms and the interfaces and control/data linkages needed to support communication among the various sub structures were established. Implementation process is simply a translation of the design abstraction into the physical realization, using the language of the target architecture.

Implementation includes all those activities that take place to convert from the old system to the new. The system may be totally new replacing an existing manual or automated system, or it may be major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet organizational requirements.

There are three types of implementations:

- Implementation of a computer system to replace a manual system.
- Implementation of a new computer system to replace an existing one.
- Implementation of a modified application to replace an existing one, using a same computer.

The common approaches for implementation are:

Parallel conversion

In parallel conversion the existing system and new system operate simultaneously until the project team is confident that the new system is working properly. the outputs from the old system continue to be distributed until the new system

has proved satisfactorily parallel conversion is a costly method because of the amount of duplication involved.

Direct Conversion

Under direct conversion method the old system is discontinued altogether and the new system becomes operational immediately. A greater risk is associated with direct conversion is no backup in the case of system fails.

Pilot Conversion

A pilot conversion would involve the changing over of the part of the system either in parallel or directly. Use of the variation of the two main methods is possible when part of the system can be treated as separate entity.

User training

After the system is implemented successfully, training of the user is one of the most important such tasks of the developer. For this purpose, user manuals are prepared and handed over to the user to operate the developed system. Thus, the users are trained to operate the developed system. In order to put new application system into use, the following activities were taken care of

- Preparation of user and system documentation
- Conducting user training with demo and hands on
- Test run for some period to ensure to smooth switching over the system

The major implementation procedures are:

- Test plan
- Training
- Equipment installation
- Conversion

Test plan

The implementation of a computer-based system requires that the test data be prepared and the system and its elements be tested in structured manner.

Training

The purpose of the training is to ensure that all the personal who are to be associated with the computer-based business system processes the necessary knowledge skills. As the system provides user friendliness, only basic training is needed.

Conversion

It is the process of performing all of the operations that result directly in the turnover of the new system to the user. Conversion has two parts:

- The creation of a conversion plan at the start of the development phase and the implementation of the plan throughout the development phase.
- The creation of a system changes over plan at the end of development phase and the implementation of the plan at the beginning of the operation phase.

6.2 DOCUMENTATION

The documentation involves collecting, organizing, and maintaining complete record of programs. The documentation deals with the system department with maximum clarity. Each and every process is explained in detail. The various table used by the system with field details are provided. The system uses various kinds of forms to produce well-structured screen formats. These forms are also documented. The output generated by the system constitutes another part. Documentation of the software provides the following:

Comments

Comments are very useful in documenting a program. It is used to explain logic of the program. It should be used to improve the quality and understand ability of the program.

It should not be redundant, incorrect or incomplete.

System Manuals

A good software system must contain standard system manuals. In this the statement is clearly defined, specifies description, detailed flowcharts, and specimen of all input forms and printed outputs.

Operation Manual

A good software package is supported with a good operation manual to ensure the smooth running of the program. The operation manual must contain the following information:

- Setup and operational details of each program.
- Loading and unloading procedures.
- Starting, running, and terminating procedures.
- List of error conditions with explanations

SYSTEM MAINTENANCE AND SECURITY

7.1 SYSTEM MAINTENANCE

The maintenance is an important activity in the life cycle of a software product. Maintenance includes all the activities after the installation of software that is performed to keep the system operational. The process of monitoring, evaluating, and

modifying of existing information systems to make required improvements may be termed as system maintenance.

The software package is delivered into the customer the following problem occurs.

- The user discovers new errors in the software
- The customer wants to upgrade his hardware, thus necessitating software changes.
- New needs of user request for the functional enhancement of software.

Maintenance is classified into four types

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

CORRECTIVE MAINTENANCE

Corrective maintenance refers to changes made to repair defects in the design, coding, or implementation of the system. Corrective maintenance is often needed for repairing, processing or performance failures or making changes because of previously uncorrected problems or false assumptions. Most corrective maintenance problems surface, they are typically urgent and need to be resolved to curtail possible interruptions in normal business activities. Here the application is tested and reviewed for the performance and the necessary changes are made. In general, modular program is easy to correct and maintain.

ADAPTIVE MAINTENANCE

Adaptive maintenance involves making changes to an information system to involve its functionality or it to different operating environment. Adaptive maintenance is usually less urgent than corrective maintenance because of business and technical changes typically occur some period of time.

The environment changes include:

- Hardware changes
- Operating system changes

PERFECTIVE MAINTENANCE

Perfective Maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. Many system professionals feel that perfective maintenance is not really the maintenance but new development.

PREVENTIVE MAINTENANCE

Preventive maintenance is regularly performed on a piece of equipment to reduce the likelihood of failure. Preventive maintenance ensures that anything of value to your organization receives consistent maintenance to avoid unexpected breakdowns and costly disruptions. In terms of the complexity of this maintenance strategy, it falls between reactive maintenance and predictive maintenance.

7.2 SYSTEM SECURITY

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively.

The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

CHECKS AND CONTROLS

When developing or acquiring software applications, it is important to ensure that the data being entered is properly checked. This Activity presents guidelines on how to check and control data entry.

Good Practices and Recommendations

The following types of checks and controls are important to have in the data entry screens in all software applications:

- Validate all fields that have ranges such as dates or amounts
- Allow the user, under privilege control, to add a parameter that is not in a lookup table on the spot without having to go to another screen.
- Allow the user to search for major tables such as citizens, projects, contractors, Etc. This should be available during deletions, modifications, printing and other system functions.
- Design screen layouts to be similar to actual vouchers. This eases data entry and requires less training for the user.
- Use clear color coding as per Windows standards: Black labels, White for enterable fields and Greyed fields for non-enterable or for system responses

- Differentiate between Info, Error and Warning messages through the proper use of buttons: Info (OK), Error (OK), and Warning (Yes, No), Choices (Yes, No, Cancel)
- Use clear and unambiguous messages
- Avoid cluttering the screen with a large number of fields. It becomes difficult to visually scan the screen and validate the data. In the case of large number of fields, it is best to use TABs or even multiple screens.
- Do not allow the system to accept to create or modify a record unless all data is validated. Many systems suffer from temporary entries that are never completed.

Security features are considered while developing the system, so as to avoid the errors and omission that may lead to serious problems. Computer system is secure against a particular threat if counter measures have been taken to reduce acceptability low level amounts of loss that the threat may be expected to cause over a given period of time. The above guidelines should be standardized across various applications to ensure that users get familiar with the look and feel of applications and hence require less training. A computer should be protected from the following three problems.

- Loss of availability
- Loss of integrity
- Loss of confidentiality

A threat to a computer system is any events that adversely affected the one or more assets or resources, which make up a system. An event can be any of the following.

- Interruption of communication
- Destruction of hardware
- Modification of programs
- Removal of programs
- Disclosure of information

There are many methods for handling a threat

- Altering the design
- Threat retention
- Threat reduction

There are many possible threats to the security and integrity of any system where more than one user is associated with the system. Software integrity has become increasingly important. The attribute measures a system to withstand attacks, both accidental and intentional on its security. Attacks can be made on all three components of software programs and documents.

7.5 SCOPE FOR FUTURE ENHANCEMENT

The future enhancement scope of a dairy management system is vast, with ongoing advancements in technology and changing agricultural needs. Here are some potential areas of improvement and future developments for dairy management systems:

1. **Integration of AI and Machine Learning:** AI and machine learning technologies can be integrated into dairy management systems to analyze large datasets, identify patterns, and make predictions. This could help in optimizing feeding plans, predicting heat cycles, detecting early signs of diseases, and improving overall farm efficiency.
2. **Precision Livestock Farming:** Precision livestock farming (PLF) involves using various technologies like sensors, wearables, and monitoring systems to collect real-time data on individual cows. Future enhancements may focus on expanding PLF capabilities, enabling better decision-making and personalized care for each animal.
3. **Blockchain for Supply Chain Transparency:** Implementing blockchain technology can enhance transparency and traceability in the dairy supply chain. Consumers and stakeholders can access immutable records of milk quality, production practices, and animal welfare measures.
4. **Internet of Things (IoT) Integration:** IoT devices and sensors can be further integrated into dairy management systems to monitor various parameters such as temperature, humidity, and air quality in barns, helping to create optimal conditions for cows and improve overall farm productivity.
5. **Automated Health Monitoring:** Advancements in health monitoring technologies may allow for the automatic detection of health issues in cows through the analysis of sensor data. Early detection of health problems can lead to timely intervention and improved animal welfare.
6. **Remote Data Accessibility:** Enhancements in cloud-based solutions and mobile applications will facilitate remote access to dairy management data. This will enable farmers to monitor and manage their dairy operations from anywhere, improving convenience and flexibility.
7. **Sustainability and Environmental Impact Tracking:** Future dairy management systems may incorporate features to track and analyze the environmental impact of dairy

farms, allowing farmers to adopt more sustainable practices and reduce their carbon footprint.

8. **Robotic Milking and Handling:** Improvements in robotic milking and handling systems can lead to increased automation on dairy farms, reducing labor requirements and enhancing overall milking efficiency.
9. **Data Analytics and Predictive Analytics:** Enhanced data analytics capabilities can provide farmers with deeper insights into their operations, helping them make informed decisions for better productivity and profitability. Predictive analytics can anticipate future trends and challenges, aiding in proactive planning.
10. **Mobile Diagnostics and Telemedicine:** The integration of mobile diagnostics and telemedicine solutions into dairy management systems may allow veterinarians to remotely assess cow health, provide recommendations, and collaborate with farmers more effectively.

Overall, the future of dairy management systems lies in leveraging emerging technologies to create more data-driven, efficient, and sustainable dairy farming practices. As these advancements are adopted, dairy farmers can expect improved herd health, increased productivity, and better economic outcomes.

CONCLUSION

8.1 CONCLUSION

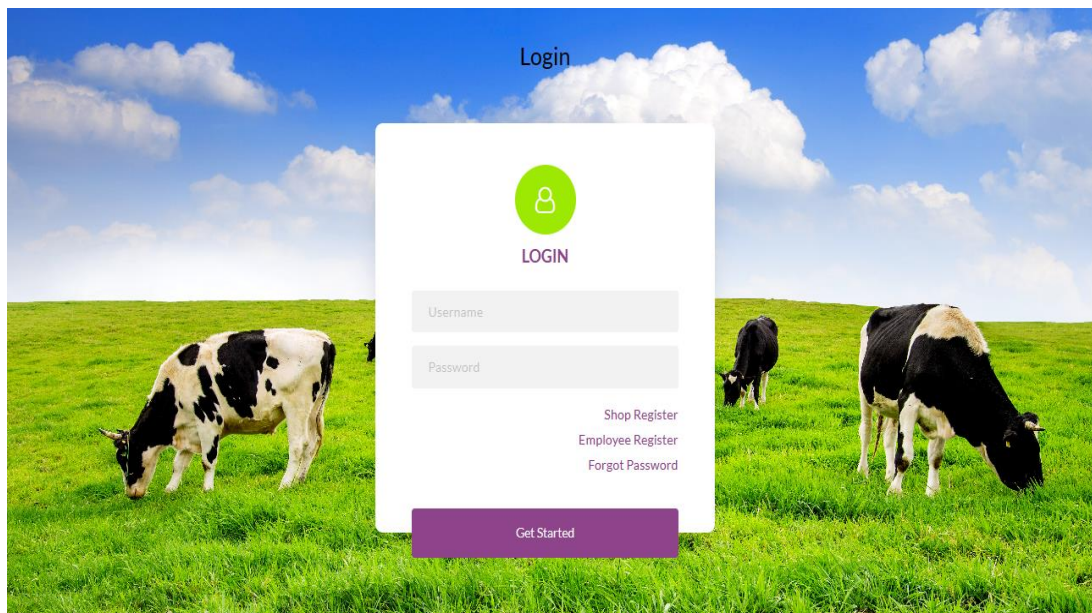
In conclusion, a well-implemented and comprehensive dairy management system is an indispensable asset for modern dairy farms. By centralizing critical information, such as individual records, milk production data, purchase and sell details, the dairy management system simplifies daily operations. This streamlining leads to enhanced efficiency and productivity on the farm. Accurate and up-to-date data empowers farmers and stakeholders to make well-informed decisions. Data-driven insights allow for proactive planning, and strategic interventions, leading to improved outcomes and profitability. The Dairy Product Management System application which deals with maintaining the inventory details of dairy product like milk, ghee, curd, and paneer. The owners of the dairy shop can solely depend on this application. Many of us want the work to be done at a faster rate. So this application will help in reducing the time for purchasing of the products. Even this application can help the small enterprises also. The user interface must be simple and easy to understand. This application will help in storing the products records like the dairy product, employee records, and customer records, purchased information in an easy and well organized manner. This will be one of the projects that will help the owners of the dairy shop to maintain the inventory and the billing process with great ease. The aim of the project is to computerize the operations of collecting the sales details from the dealer and to book order for their next dispatching. The modules are in here Admin, User, Shop, Farmer, Employees, Delivery boy. They are involved in this system. Delivery boy deliver the product from admin to user and shop. Shops buy the bulk load of the product and sell to the other peoples.

APPENDIX

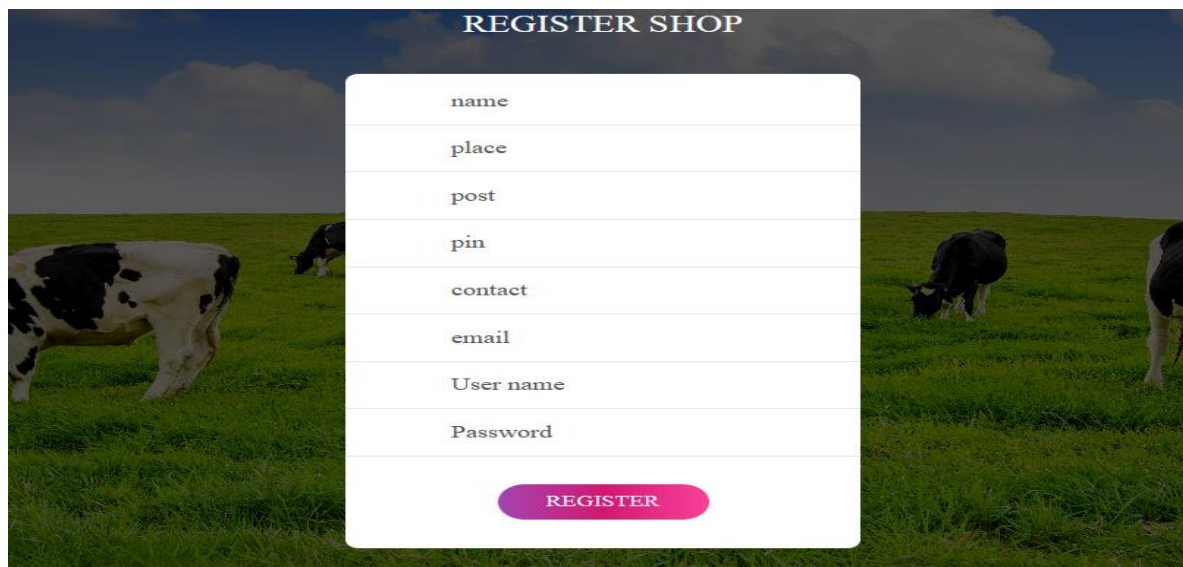
9.1 SCREENSHOT

Web part sample screenshots:

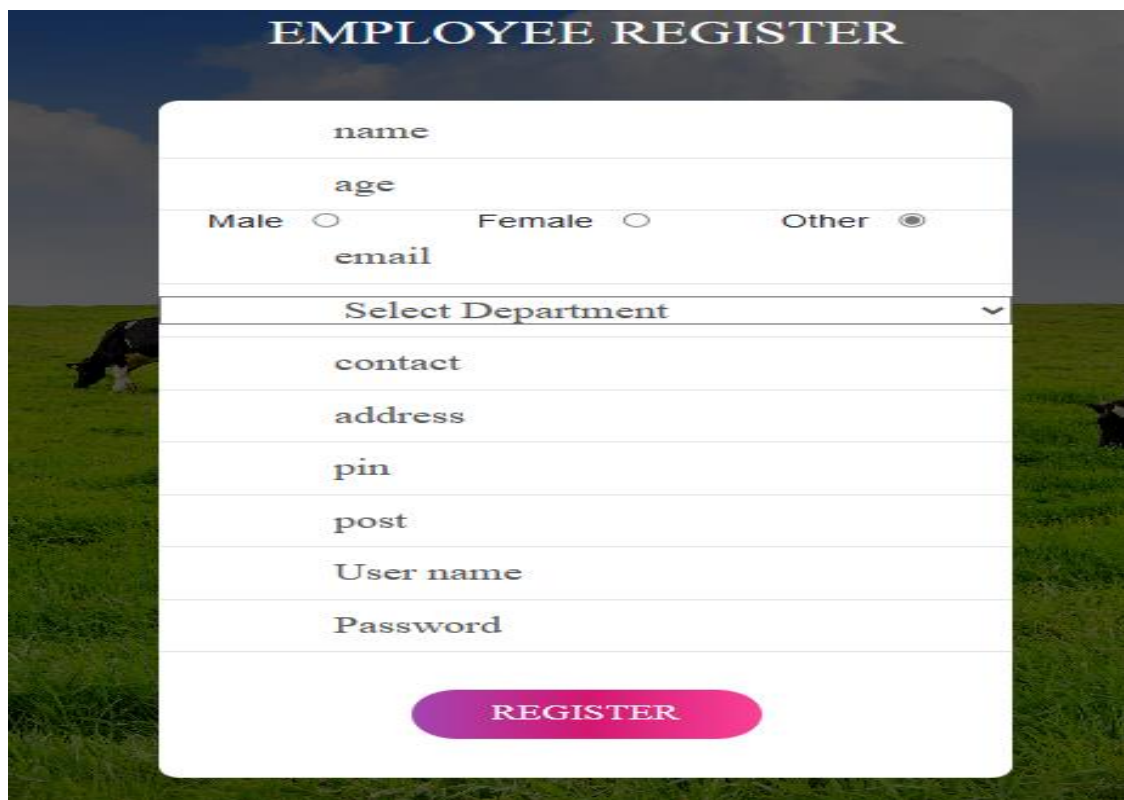
Login page



Shop Registration page



Employee Registration Page



The image shows a web form titled "EMPLOYEE REGISTER" overlaid on a background image of a green field with cows. The form is a white rounded rectangle with the following fields and controls:

- name
- age
- Gender selection: Male ☐, Female ☐, Other ☒
- email
- Select Department (dropdown menu with a downward arrow)
- contact
- address
- pin
- post
- User name
- Password
- REGISTER button (pink rounded rectangle)

ADMIN HOME PAGE

ADMIN HOME									
DELIVERY BOY	USERS	VERIFY SHOP	EMPLOYEE ▾	FARMER ▾	PRODUCT	NOTIFICATION	ORDER	COMPLAINT	VIEW FEEDBACK
								MILK ▾	LOGOUT




View delivery boy

manage deliveryboy				
name	age	email	lisence	
Roshan	21	Roshan@gmail.com	storage_emulated_0_DCIM_Screenshots_20230616_073121.jpg	780
Safa	22	smonusafeer113@gmail.com	storage_emulated_0_Android_media_com.whatsapp.w4b_WhatsApp_Business_Media_WhatsApp_Business_Images_IMG-20230629-WA0000.jpg	812
safeer	18	smonusafeer113@gmail.com	storage_emulated_0_Android_media_com.whatsapp.w4b_WhatsApp_Business_Media_WhatsApp_Business_Images_IMG-20230718-WA0000.jpg	812
Shabeeb	24	shabi123@gmail.com	storage_emulated_0_Android_media_com.whatsapp.w4b_WhatsApp_Business_Media_WhatsApp_Business_Images_IMG-20230718-WA0000.jpg	973

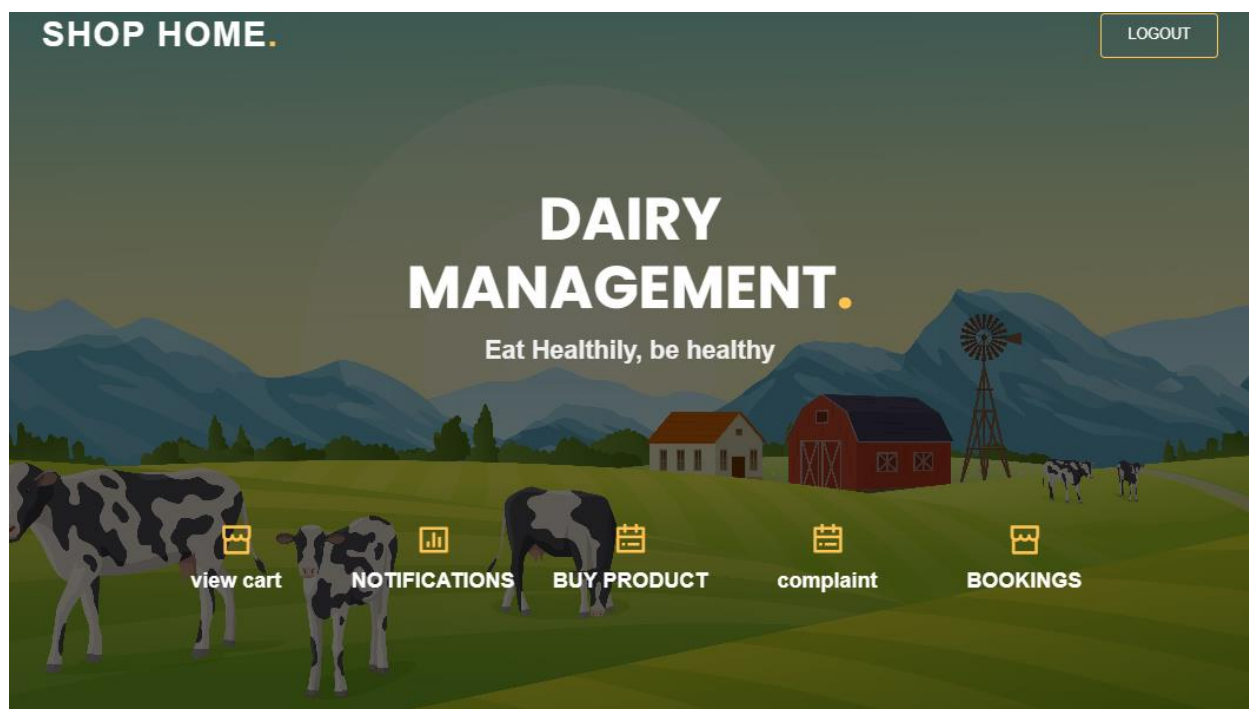
View and verify shop

view and verify shop						
shop	email	phone	place	pin	post	status
KH bakery	khbakes@gmail.com	9876543210	kunnappally	679322	kunnappally	accepted
jamjoom	jmjm@email.com	9876543342	perinthalmanna	679322	perinthalma	accepted
rinshabakery	rinshabakery@email.Com	9867809830	kunnappally	679322	kunnappally	accepted


Add and manage Products

view/manage product						
product	price	image	stock	details	edit/delete	
PANEER	80.0		73	GOOD HEALTHY	edit	delete
Addnew						

Shop home page




View cart page

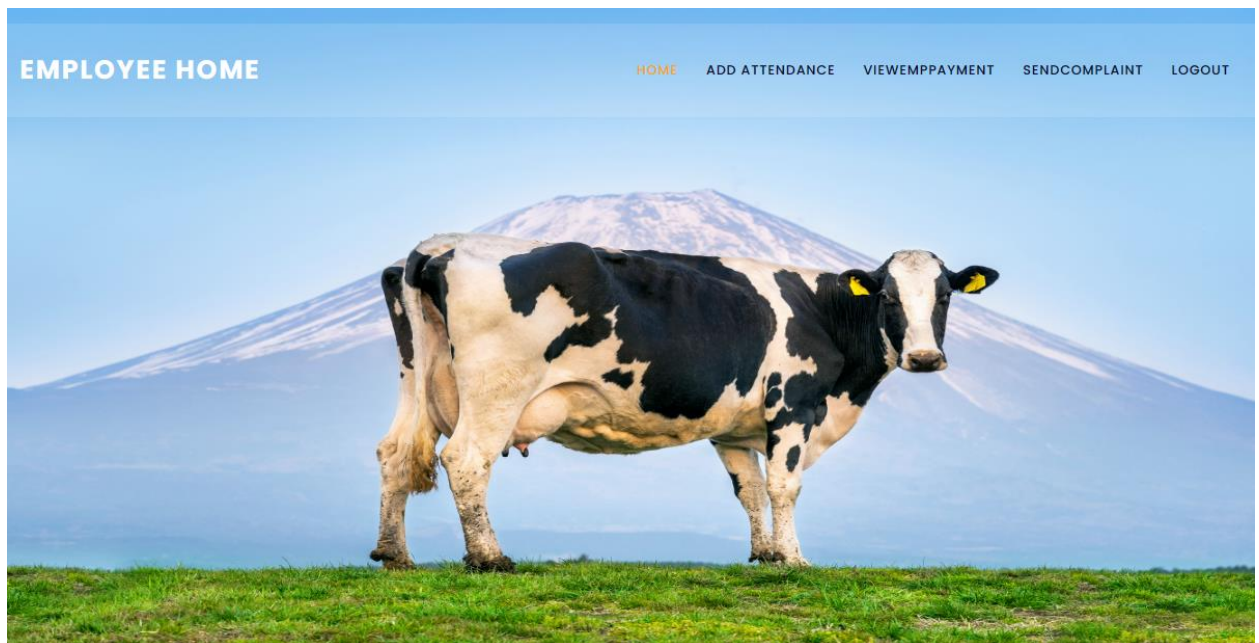
view cart			
product	image	quantiy	
PANEER		<div><div></div>4<div></div></div>	delete
total	320.0		
<div>order</div>			

View notifications page

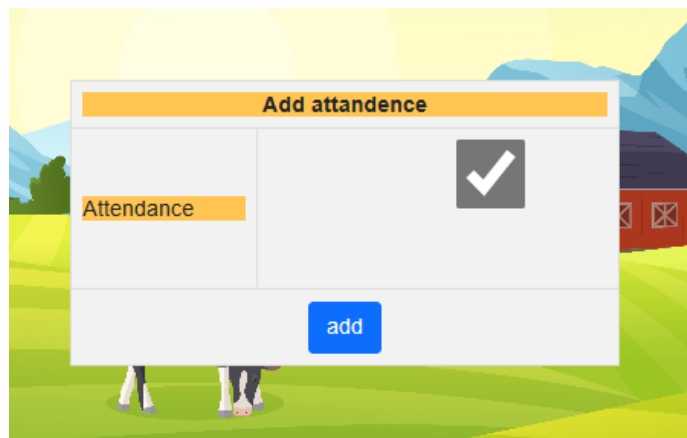
View Notification	Date
new	2023-07-09
hi	2023-07-11
hi my family new shop is comimng soon	2023-07-21

Buy product page

View product and Book				
Product name	Price	Image	Stock	
PANEER	80.0		73	buy or book

Employee home page

Add attendance page



View payment page

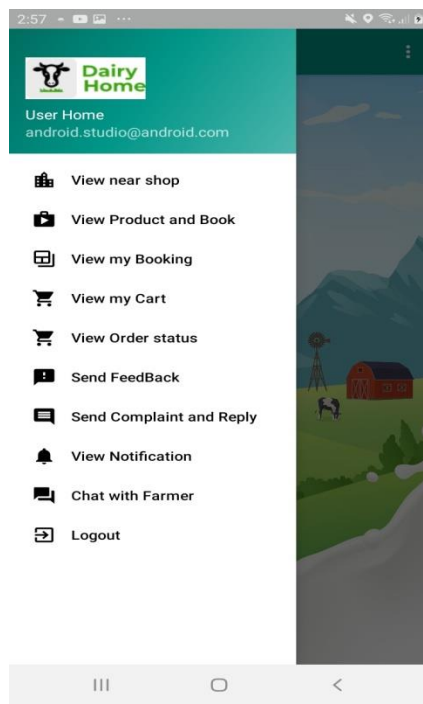
View payment			
Amount	Details	Type	Date
5000.0	salary	ONLINE PAYMENT	2023-07-28
100.0	insentive	CASH ON HAND	2023-07-28

Android part sample screenshots:

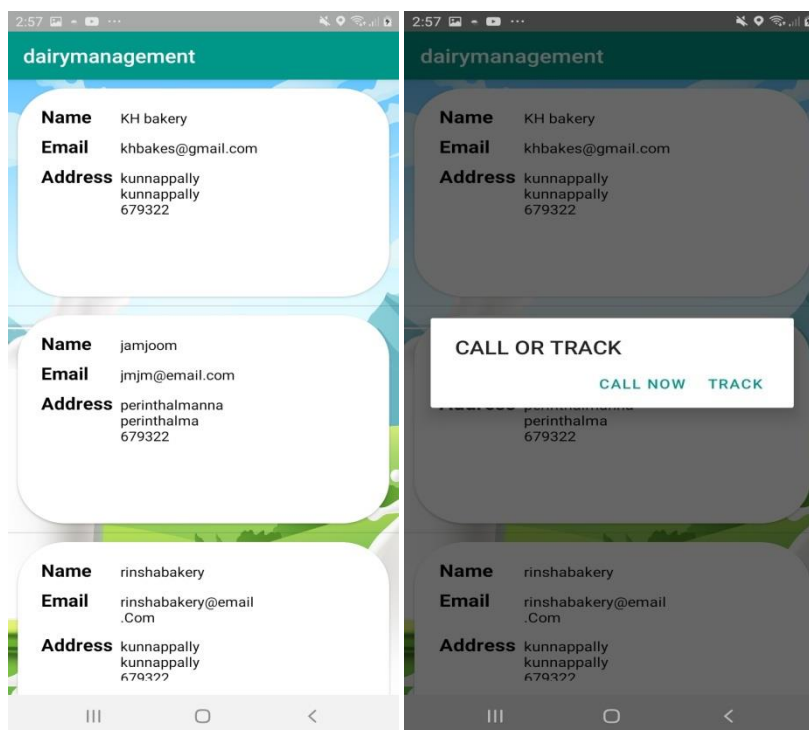
Login

Registration pages

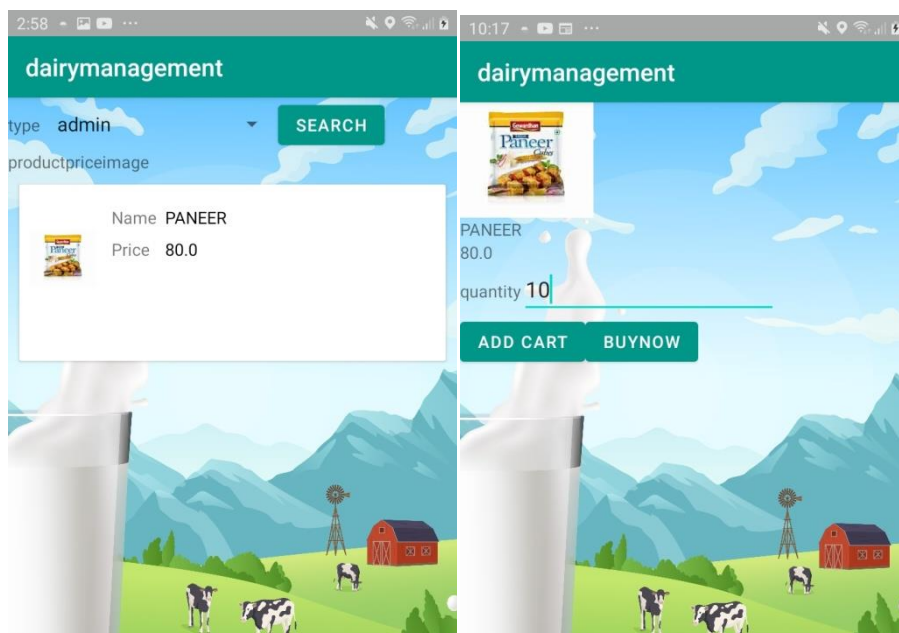
User Home page



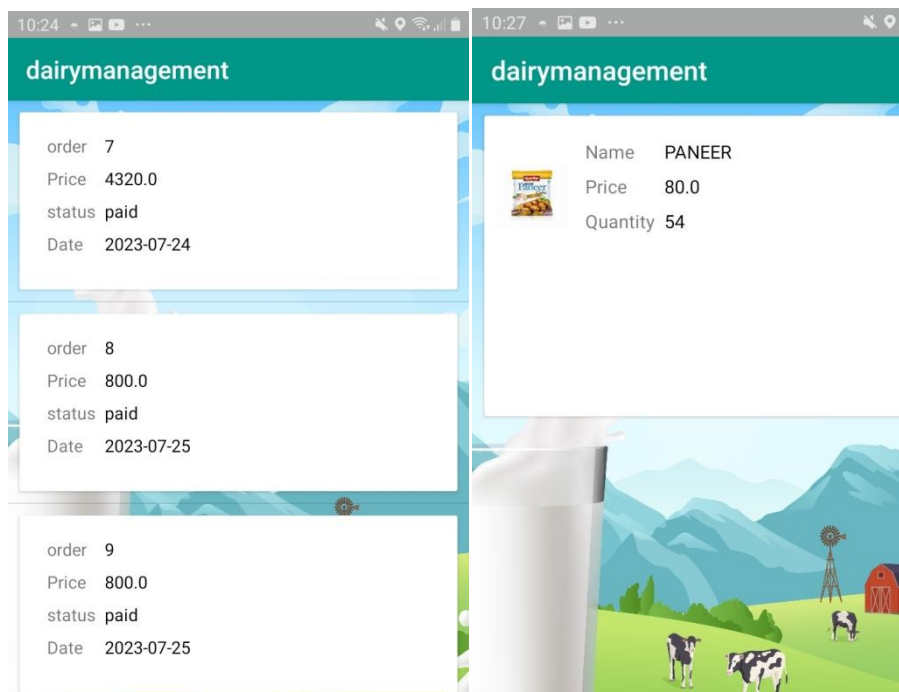
View near shop



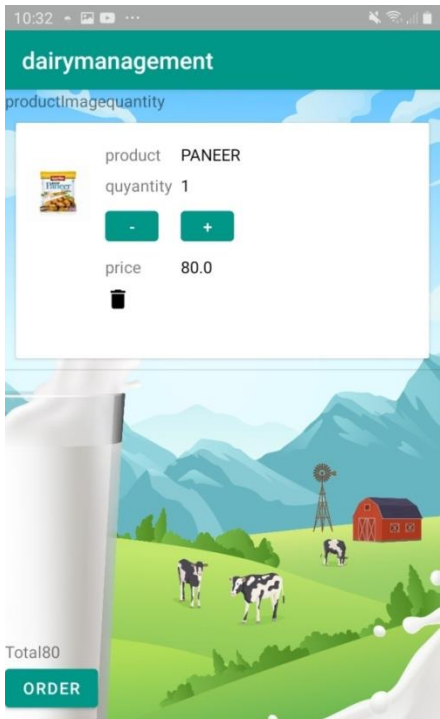
View product and book



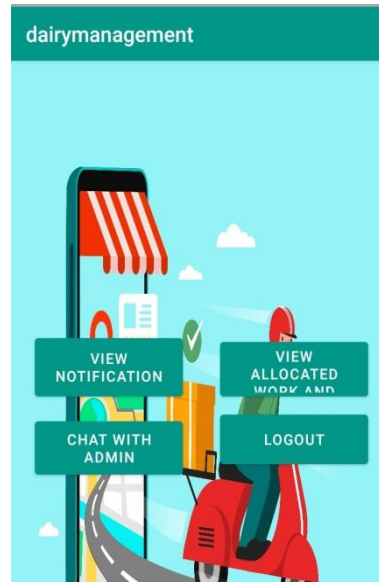
View bookings page



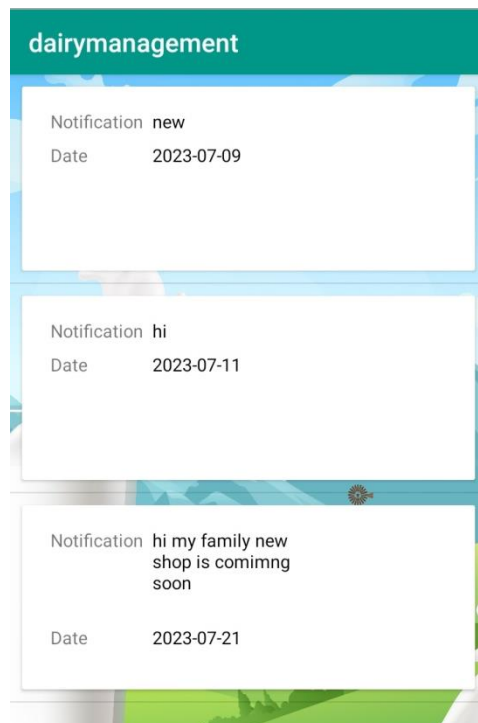
View cart page



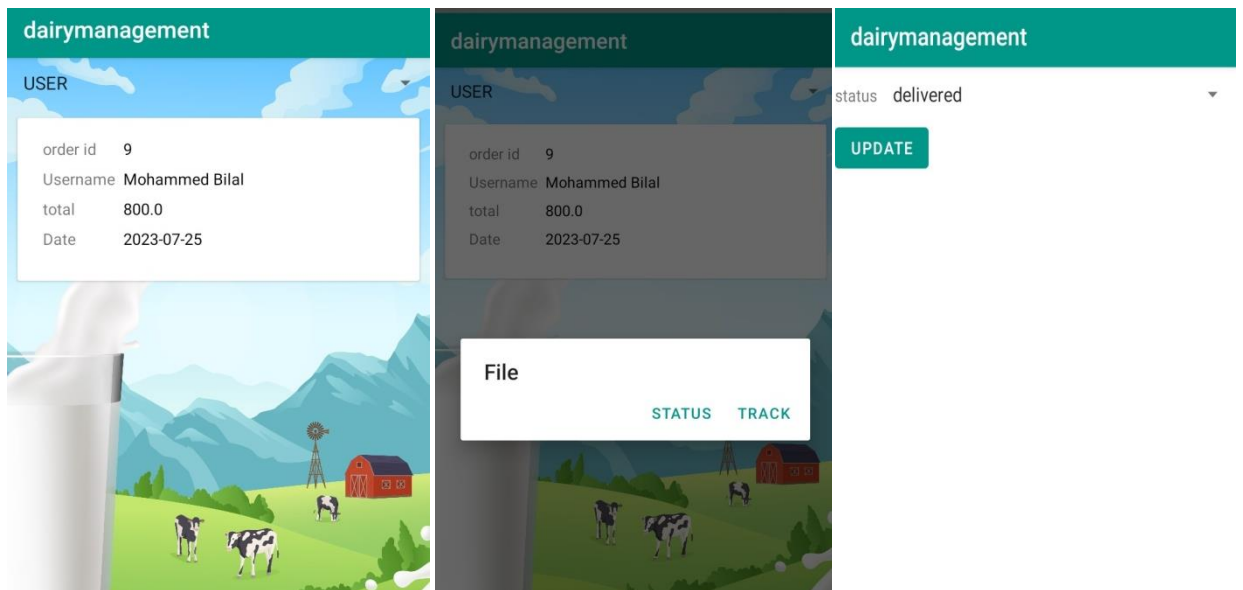
Delivery boy Home page



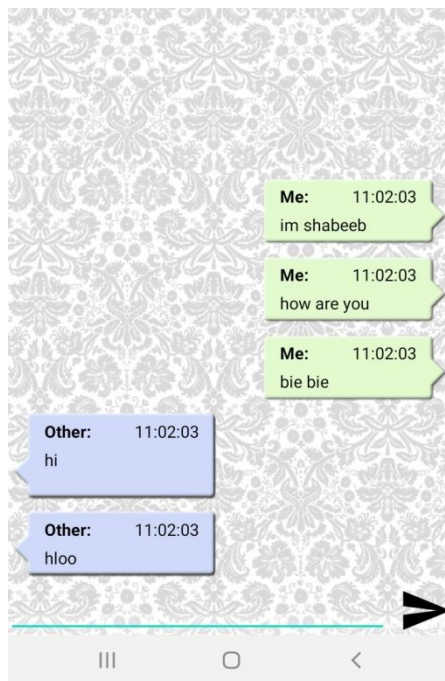
View Notification page



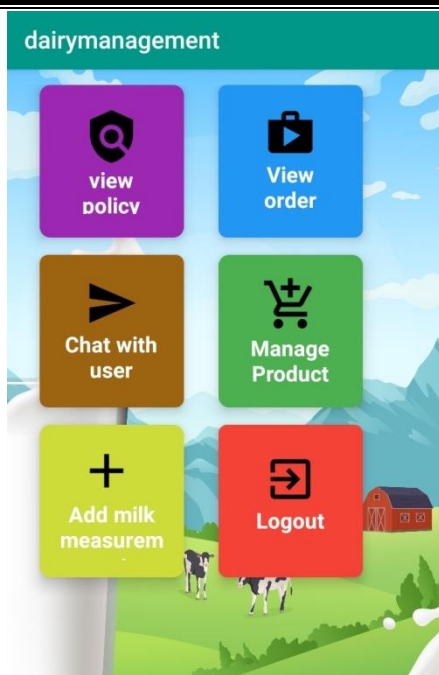
View Allocated Work & Track page



Chat with Admin



Farmer Home page



View Policy page



Chat with user page

dairymanagement

jaseela

Mohammed Bilal

Me: 11:02:03
im shabeeb

Me: 11:02:03
how are you

Me: 11:02:03
bie bie

Other: 11:02:03
hi

Other: 11:02:03
hloo



View and Manage Product

dairymanagement

ADD



Name calf
Price 5000.0
Quantity 1

dairymanagement

product name

PRODUCT PRICE

IMAGE

CHOOSE

STOCK

DETAILS

ADD

Sample code

```

import os
import smtplib
from email.mime.text import MIMEText

from flask import *
from werkzeug.utils import secure_filename

from datetime import timedelta, datetime

from src.dbconnection import *
app = Flask(__name__)
app.secret_key="hghg"

@app.route("/", methods=['GET', 'post'])
def home():
    return render_template("loginindex.html")
@app.route("/adminhome", methods=["GET", "POST"])
def adminhome():
    q = "SELECT * FROM `deliveryboy` JOIN `login` ON `login`.`lid`=`deliveryboy`.`lid` "
    res = selectall(q)
    for i in res:
        q1 = "SELECT COUNT(`status`) AS total FROM `chat` WHERE `status`='pending' AND `to_id`=%s AND `from_id`=%s"
        rs = selectone(q1, (session['lid'], i['lid']))
        print(rs, "DDDDDDDDDDDDDDDDDDDD")
        try:
            if int(rs['total']) != 0:
                i['c'] = "(" + str(int(rs['total'])) + ")"
            session['cnt'] = i['c']
            print(session['cnt'], "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB")
            print(i['c'], "JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ")
        else:
            i['c'] = ""
    except:
        i['c'] = ""
    return render_template("adminindex.html")

@app.route('/login', methods=['GET', 'POST'])
def login():

```

```

        username = request.form["textfield"]
        password = request.form["textfield2"]
        qry="select * from login where username =%s and password
        =%s"
        val = (username,password)
        s=selectone(qry,val)
        print(s)
        if s is None:
            return
            '''<script>alert("invalid");window.location='/</script>'''
        elif s['type']=="admin":
            session['lid']=s['lid']
            print(s,"kkkkkkkkkkkkkkkkkkkkkk")
            return '''<script>alert("welcome
            adminhome");window.location='/adminhome'</script>'''
        elif s['type'] == "shop":
            session['lid']=s['lid']

            return '''<script>alert("welcome
            shophome");window.location='/shophome'</script>'''
        elif s['type']=="employee":
            session['lid']=s['lid']

            return '''<script>alert("welcome
            employeeshome");window.location='/employeeshome'</script>'''
        else:
            return
            '''<script>alert("invalid");window.location='/</script>'''

@app.route("/addpolicy",methods=['get','post'])
def addpolicy():
    return render_template("admin/Add govnt policy.html")

@app.route("/addgovpolicy",methods=['GET','POST'])
def govntpolicy():
    policy = request.form['select']
    details = request.form['textarea']
    formore=request.form['textarea2']
    qry = "insert into policy values (null,%s,%s,%s)"
    val=(policy,details,formore)
    iud(qry,val)
    return '''<script>alert("govnt policy added
    successfully"),window.location='viewpolicy'</script>'''

```

```

@app.route("/adddeliveryboy",methods=['GET','POST'])
def adddeliveryboy():
    name = request.form['textfield']
    age = request.form['textfield2']
    mail = request.form['textfield3']
    liscence = request.files['file']
    nnn=secure_filename(liscence.filename)
    liscence.save(os.path.join('static/dimg',nnn))

    contact = request.form['textfield4']
    address =request.form['textarea']
    username = request.form['textfield5']
    password = request.form['textfield6']
    q="insert into login values (null,%s,%s,'delivery boy')"
    val=(username,password)
    id=iud(q,val)

    qry = "insert into deliveryboy values
    (null,%s,%s,%s,%s,%s,%s,%s)"
    val = (str(id),name,age,mail,nnn,contact,address)
    iud(qry,val)

    return '''<script>alert("deliveryboy added
    successfully");window.location='/adminhome'</script>'''
@app.route("/adddelboy",methods=['get','post'])
def adddelboy():
    return render_template("admin/adddeliveryboy.html")

@app.route("/emp",methods=['get','post'])
def emp():
    return render_template("employee/regemployee.html")

@app.route("/addproduct",methods=['GET','POST'])
def addproduct():
    return render_template("admin/addproduct.html")

@app.route("/updateprod",methods=['get','post'])
def updateprod():
    try:
        product = request.form['textfield']
        price = request.form['textfield2']
        image = request.files['file']
        img = secure_filename(image.filename)
        image.save(os.path.join('static/product', img))
        stock = request.form['textfield3']
        qry = "UPDATE `product` SET
        `product`=%s,`price`=%s,`image`=%s,`stock`=%s WHERE

```



```

`productid`=%s"
val = (product, price, img,stock, session['ppid'])
    iud(qry,val)
return '''<script>alert("updated
successfully");window.location='/viewproduct'</script>'''
except Exception as e:
    product = request.form['textfield']
    price = request.form['textfield2']
    stock = request.form['textfield3']
    qry = "UPDATE `product` SET
`product`=%s,`price`=%s,`stock`=%s WHERE `productid`=%s"
val = (product, price, stock,session['ppid'])
    iud(qry, val)
return '''<script>alert("updated
successfully");window.location='/viewproduct'</script>'''

@app.route("/editprod",methods=['get','post'])
def editprod():
    id = request.args.get('id')
    session['ppid']=id
    q = "SELECT * FROM `product` WHERE `productid`=%s"
    re = selectone(q, id)
    return render_template("admin/editproduct.html",val=re)

@app.route("/deleteprod",methods=['get','post'])
def deleteprod():
    id=request.args.get('id')

    q="DELETE FROM `product` WHERE `productid` =%s"
    iud(q,id)
    return '''<script>alert("deleted
successfully");window.location='/viewproduct'</script>'''

@app.route("/milkmeasurement",methods=['get','post'])
def milkmeasurement():
    q="SELECT * FROM `farmer`"
    res=selectall(q)
    return render_template("admin/milkmeasurement.html",val=res)

app.run(debug=True)

```


BIBLIOGRAPHY

10.1 BIBLIOGRAPHY

Textbooks:

- Sommerville: Software Engineering
- Marvin F. Korth: Database System Concept
- Pressman: Software Engineering
- Marvin Gore: System Analysis and Design

Websites:

- <https://www.github.com/>
- <https://w3school.com/>
- <https://www.python.org/>
- <https://www.google.com/>
- <https://coderbyte.com/>