

## INTRODUCTION

The project “**FARMING ASSISTANT**” is an web application. In which different farmers from different locations to can post their raw food products in this website. Only registered farmers will be capable of selling their product through this website. Suppliers and other users can buy the cultivated raw products from the available farmers.

This site allows for good farmer and Customer communication. It allows farmers to login and communicate to respective Customers and Organizations. When organisation publish an advertisement or offer, the respective farmers get notified via SMS message. The farmers may also submit their gradiences and complaints to respective authorities using their farmer login on a separate complaints page and authorities will get access to that page regularly using their login id and passwords.

We have an android application also along with the web. Farmers can have the android app. It provides necessary notifications in real-time. So that he can act upon necessary. Our system also has a section to find the best farmer monthly. We use reviews and ratings of users to find out it. Only users who brought from the farmer can review and rate that farmer. So that frauds can be controlled. We use review mining technique to find the mood of the review.

## SYSTEM ENVIRONMENT

### HARDWARE REQUIREMENTS:

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

- Processor : Intel Pentium Quad Core
- Ram : 2GB
- Monitor : 17" LCD
- Mouse : INTEX Mouse OptiNeo
- Keyboard : Windows Compatible
- Storage : 500 GB Hard Disk

### SOFTWARE REQUIREMENTS:

One of the most difficult task is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

- Operating System : Windows 8
- Front End : C#.Net, android Sdk
- Back End : SQL server 2008
- IDE : Microsoft Visual Studio 2010, Eclipse

## TECHNOLOGY

### Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[6] C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

Intellisense is supported for the included languages, as well as for XML and for cascading style sheets and javascript when developing sites and web applications. Auto complete suggestions are popped up in a modeless list box, overlaid on top of the code editor. In visual studio 2008 onwards, it can be made temporarily to see the code obstructed by it. The code editor is used for all supported languages.

### C#.NET

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language

infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.

## Eclipse IDE

Eclipse is an Integrated Development Environment (IDE) that is open source and maintained by many software development leaders like IBM and Borland. The idea behind the Eclipse platform is to provide a common environment that companies can modify and customize by creating plug-ins. Plug-ins are created to do some specific tasks that a work group might need while developing a software product. The plug-ins and modification of the Eclipse platform can later be distributed royalty free.

In Computer programming **Eclipse** is a multi-language (IDE) comprising a base work space and an extensible plug in system for customizing the environment. It is written mostly in java. It can be used to develop applications in Java and, by means of various plug-ins, other programming language including Ada, C, C++, COBOL, Fortran, Haskell Java Script etc. It can also be used to develop packages for the software Mathematical Development environments include the Eclipse

Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM visual age The Eclipse (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

## SYSTEM ANALYSIS

Analysis report contains a detailed study of existing system like how it is working, what are the drawbacks of the system, etc. From that study we can understand the need of new system. In this report we also include the details of new system such as its functionalities, subsystems, functional descriptions, etc. we also tried to find any risk associated with the system and study the feasibility of the system. We also include a project scheduling for the entire project through analysis report.

### EXISTING SYSTEM:

Existing system is the present system, now all the processes are done manually. The existing system is not computerized so it consists of so many limitations. Agriculture is the backbone of Indian economy. Many farmers are not capable of travelling long distance in order to sell their products effectively at markets. There are many chances for the decay of products when it's kept for a longer time. So he should find an alternative system to sell their products effectively. Thus we propose a new system in order to solve these entire problems.

### PROPOSED SYSTEM

To overcome the drawback of the existing system, we need to develop an application for farmers, which is more easily accessible to public. Our system will help its users by providing accurate data with shortest time. The system is GUI (graphical user interface) based, so it is more user friendly. This project helps the farmers very effectively by giving a direct contact with the suppliers so that he can make use of this site to sell their products wisely. The farmers can update their stock details and can provide a contact number for user or suppliers to approach them for their products. This can be an effective selling-buying system.

## **FEASIBILITY STUDY**

The data collection that occurs during the preliminary investigation examines system feasibility. Feasibility study is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative.

When the positive impact dominates the negative, then the system is considered feasibility. There tests of feasibility are studied technical, operational and economic feasibility.

### **TECHNICAL FEASIBILITY**

Evaluating the technical feasibility is the trickiest part of the feasibility study. It is that whether the available resource are enough to carry out the project i.e., both hardware and software configuration and other equipment are in hand. The proposed system system both hardware and software requirements have been specified in system configuration. And it is sure that this project does not need resources that were not available. Thus, the proposed system is technically feasible.

### **ECONOMICAL FEASIBILITY**

The project is economically feasible as the back end, namely sql server 2008 is already in organisation .and moreover the platform being .net accessing of C#.net is available. Hence the project counts with no extra cost and its benefits outlays the investment.

### **OPERATIONAL FEASIBILITY**

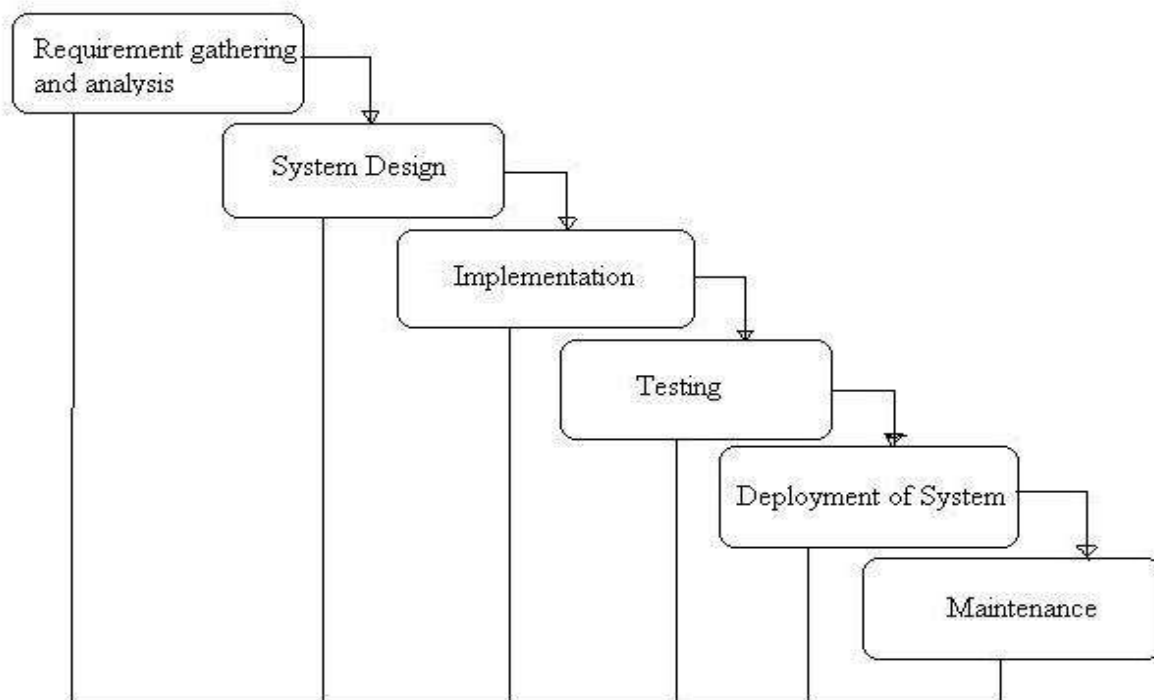
Operational feasibility access the extend to which the required software system performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources and involves visualizing whether the project will operate after it is developed and be operative once it is installed.

## METHODOLOGY

A system development methodology refers to the framework that is used to structure, plan and control the process of developing an information system.

### WATERFALL MODEL

General Overview of "Waterfall Model"



Waterfall approach was first SDLC model to be used widely in software engineering to ensure success of the project in this model; the whole process is Phase acts as the input for the next phase sequentially.

Sequential phases in waterfall model are:

- **System/information engineering:** system engineering is the process of collecting information at system level where as information engineering include collecting information where all the decision are taken.
- **Design:** it plans solution of problem that specified in the SRS document.

- **Coding:** translate design of system in to given programming language.
- **Testing:** it is a major quality control which detects defects. Its goal is to uncover requirement, design and coding errors in programs.
- **Implementation & maintenance:** this phase focuses on modifying software, correcting errors and improving performance of software.

**ADVANTAGES:**

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and not dynamic,
- There is no unambiguous requirement.
- Simple approach



## SYSTEM DESIGN

System design transforms a logical representation what the system is required to do into the physical specification. The specifications are converted into a physical reality during the development. Design forms a blue-print of the system and adds how the components relate to each other. The design phase proceeds accordingly to an ordinary sequence beginning with review and assignment of task and ending with package design. Design phase is the life cycle phase in which the detailed design of the system selected in the study phase is accomplished. A smooth transition from the study phase to design is necessary because the design phase continues the activities in the earlier phase. The first step in the design phase is to design the database and then input and output within predefined guidelines.

## ARCHITECTURAL DESIGN

### 1. ADMIN

Overall management of web part is responsibility of admin. The admin controls overall management of the institute, which includes are:

- Login
- Add Seed/Crop/Fertilizer
- View Device Accept/Reject
- View Device Request
- View Rating and Comment
- Best Farmer
- Approve/Reject Farmer
- Approve/Reject Organisation

### 2. FARMER

The Farmer manages are:

- Registration
- Login
- Add Products & Manage Products
- View Payment
- View Device Details
- Request for Devices
- View Farmer Rating and Comments
- View Queries and Post Response

- View Organisation Ads
- Send Device Request
- Post Farmer Land Details
- View Land [ From User ]
- View Alerts [ Products, Fertilizers ]
- View Seeds/Fertilizers/Crops

### **3. USER**

- Registration
- Login
- Approve/Reject Farmer
- View Seeds/Fertilizers/Crops
- View Acknowledgement
- View Farmer Product
- Request for Farmer Product
- Post Query & Response
- Post Comment
- Add Rating
- Post Land details & View Request
- View Alerts [ Product / Fertilizer ]

### **4. ORGANISATION**

- Registration
- Login
- Post Ads on Device , Fertilizers
- View Request
- Post Discounts
- View Device Request, Accept or Reject
- View Payment

## DATA FLOW DIAGRAM

As the name suggests DFD is a representation of the data flow within the system. It includes processes, data stores and external interfaces to the system. External interfaces may be treated as external entities. DFD's can be exploded to sub-processes. Any two entities/data stores must be connected to each other by some process. The processes cannot be directly connected to each other. The following are the four major components of the DFD:

- Process
- External entity
- Data store

It is also known as the bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out in these data and the output data generated by the system. The main reason why this DFD technique is so popular is probably because of the facts that DFD is very simple formalism. It is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among these systems. Starting with a set of high-level functions that a system performance of DFD model in hierarchically it represents various sub functions. The data flow diagramming technique also follows a simple a simple set of intuitive concepts and rules.

- **Process:** -

Process shows the work of the systems. Each process has one or more data inputs and produce one or more data outputs. Processes are represented by rounded rectangles in Data Flow Diagrams. Each process has a unique name and number. This name and number appears inside the rectangle that represents the process in a Data Flow Diagram

- **Data Stores:** -

A data store is a repository of data. Processes can enter data, into a store or retrieve the data from the data store. Each data has a unique name.

- **Data Flows:** -

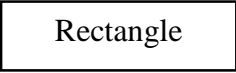
Data flows show the passage of data in the system and are represented by lines joining system components. An arrow indicates the direction of flow and the line is labeled by name of the data flow.

- **External Entity: -**

External entities are outside the system but they either supply input data into the system or use another systems output. They are entities on which the designer has control. They may be an organizations customer or other bodies with which the system interacts. External entities that supply data into the system are sometimes called sinks. These are represented by rectangles in the Data Flow Diagram.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figure, usually circles, which are called nodes.

Basic data flow diagram symbols are;



Rectangle

Represent an external entity that is the source or destination of data within the system. Each external entity is represented by a meaningful and unique name



Circles

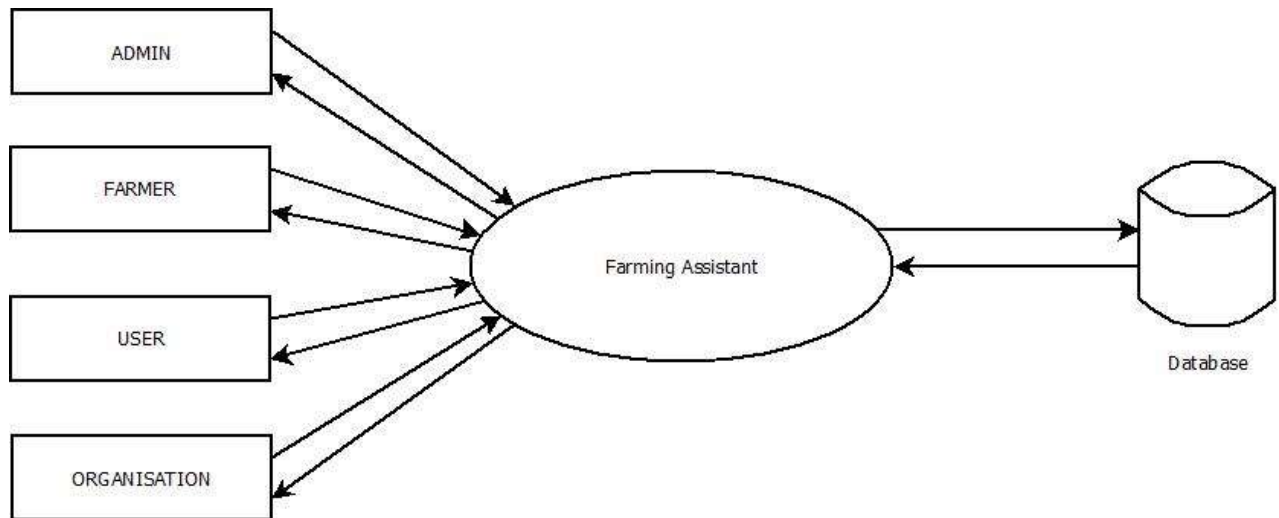
Represent processes that sow transformation or manipulation of data within the system

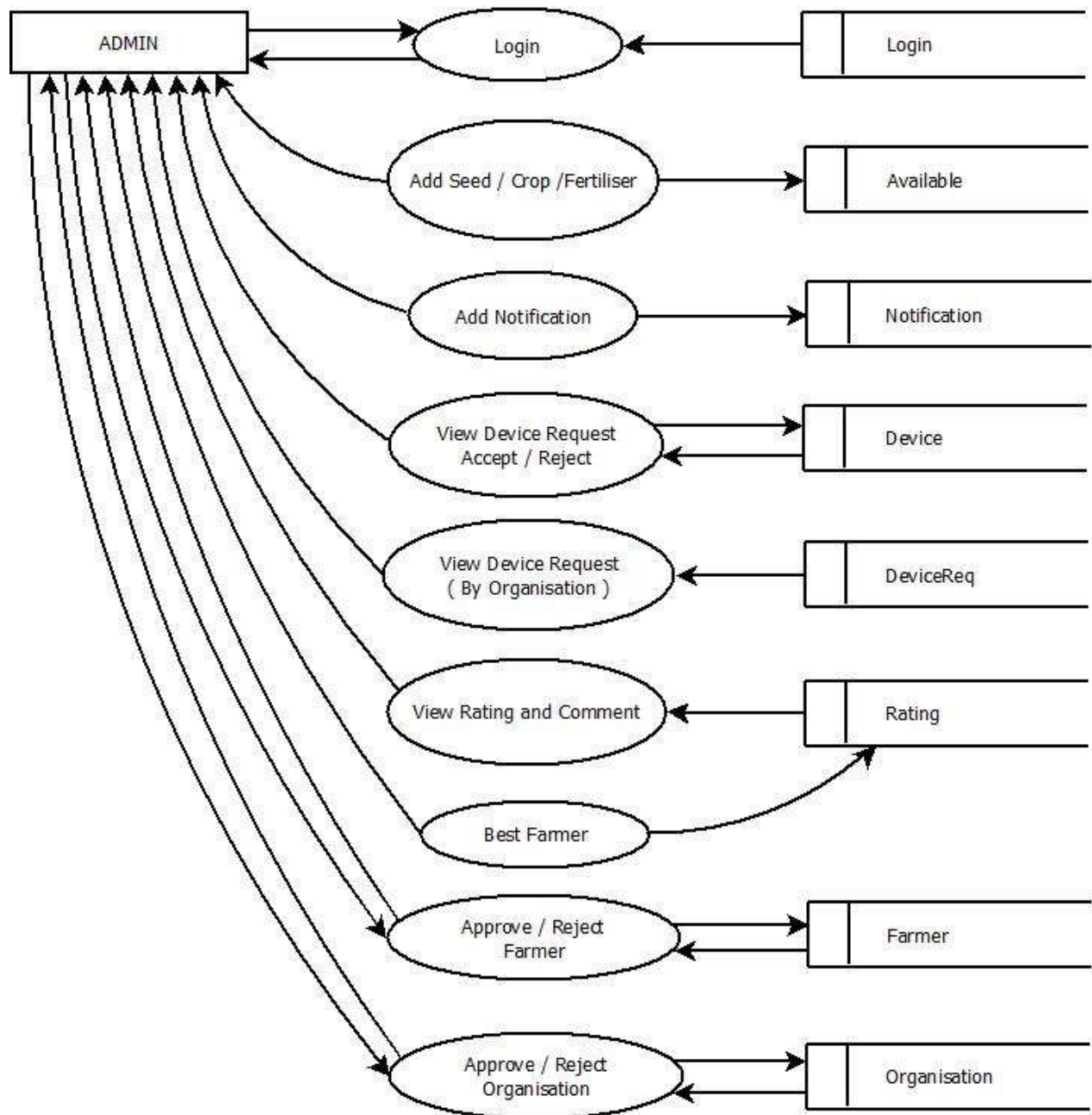


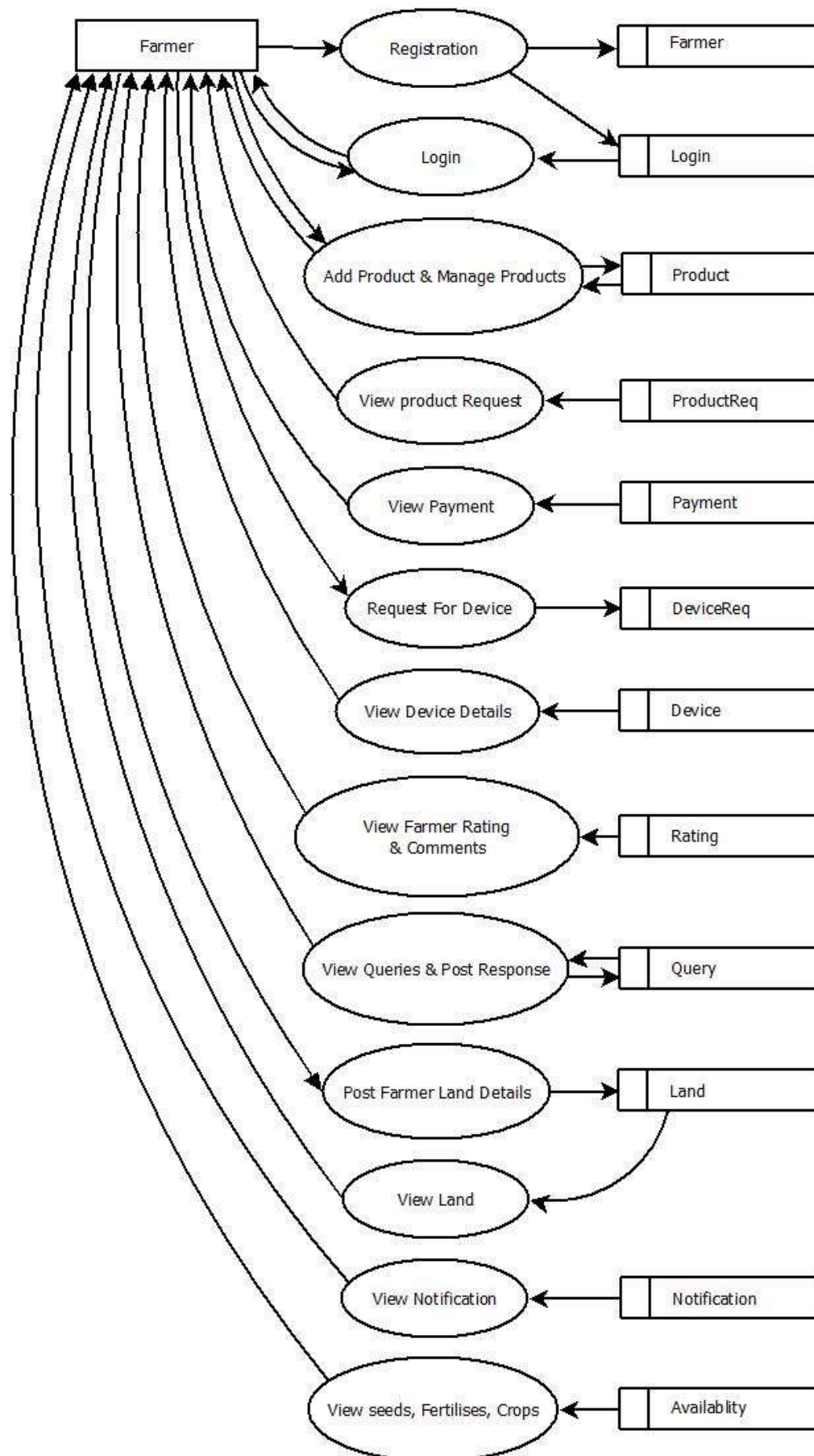
Represent data stores that indicate the place for storing information within the system



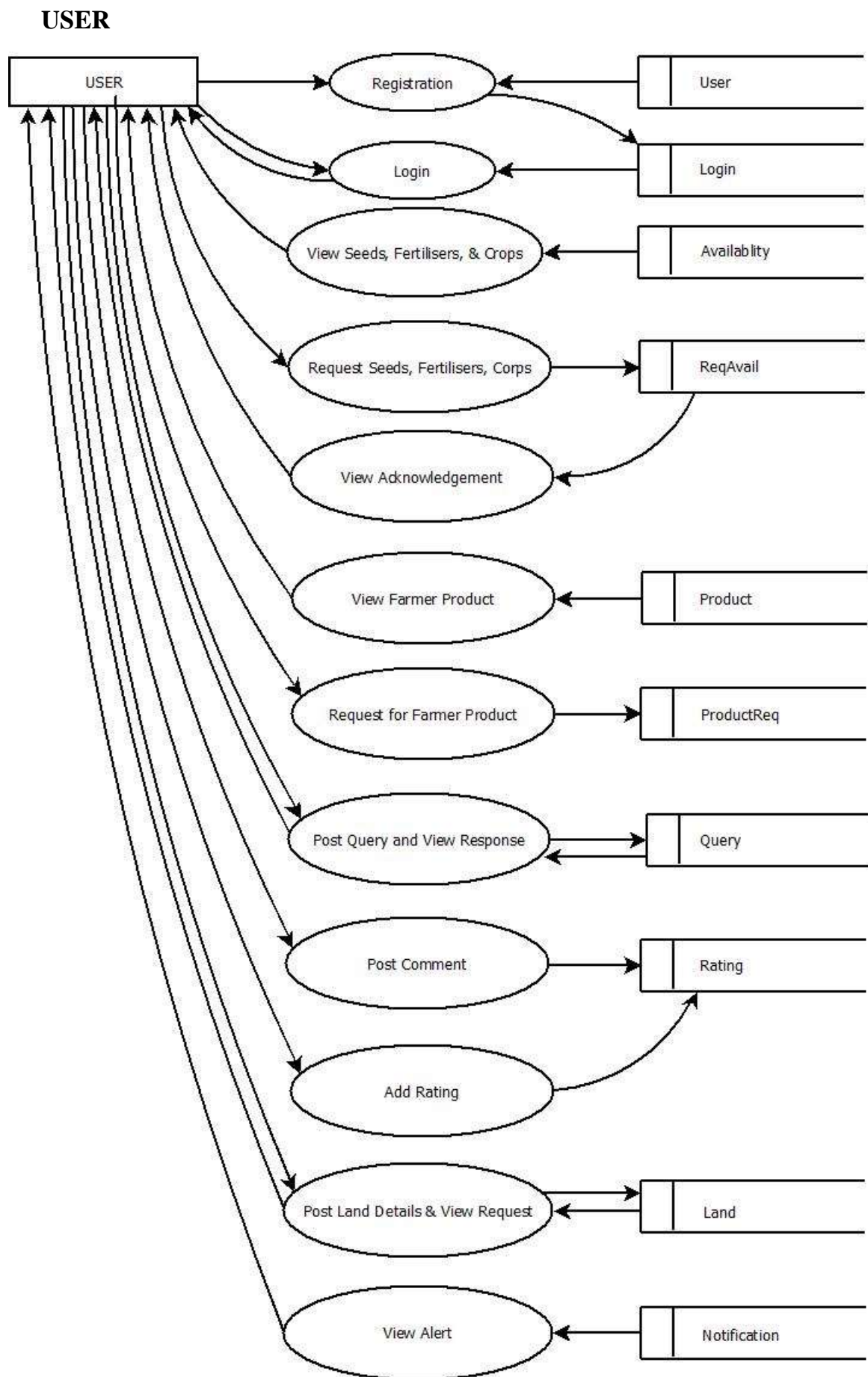
They are used to represent data flows that show the movement of data from its source to destination within the system.

**LEVEL 0 (CONTEXT DESIGN)**

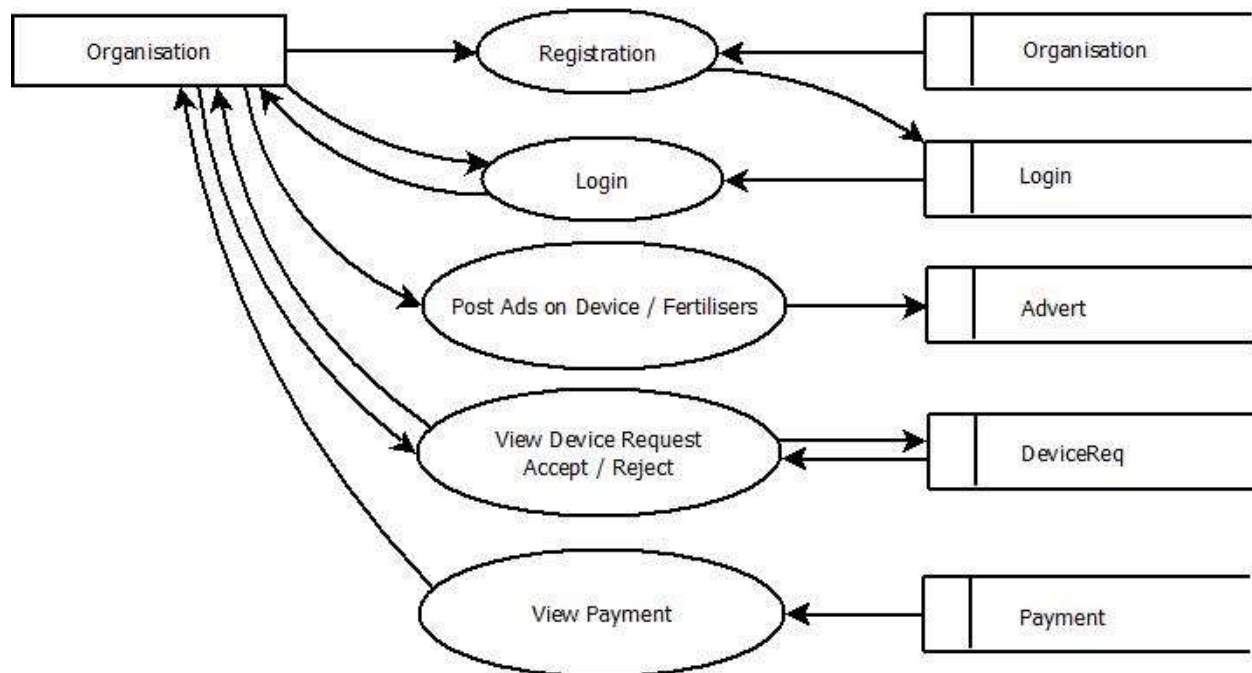
**LEVEL 1****ADMIN**

**FARMER**





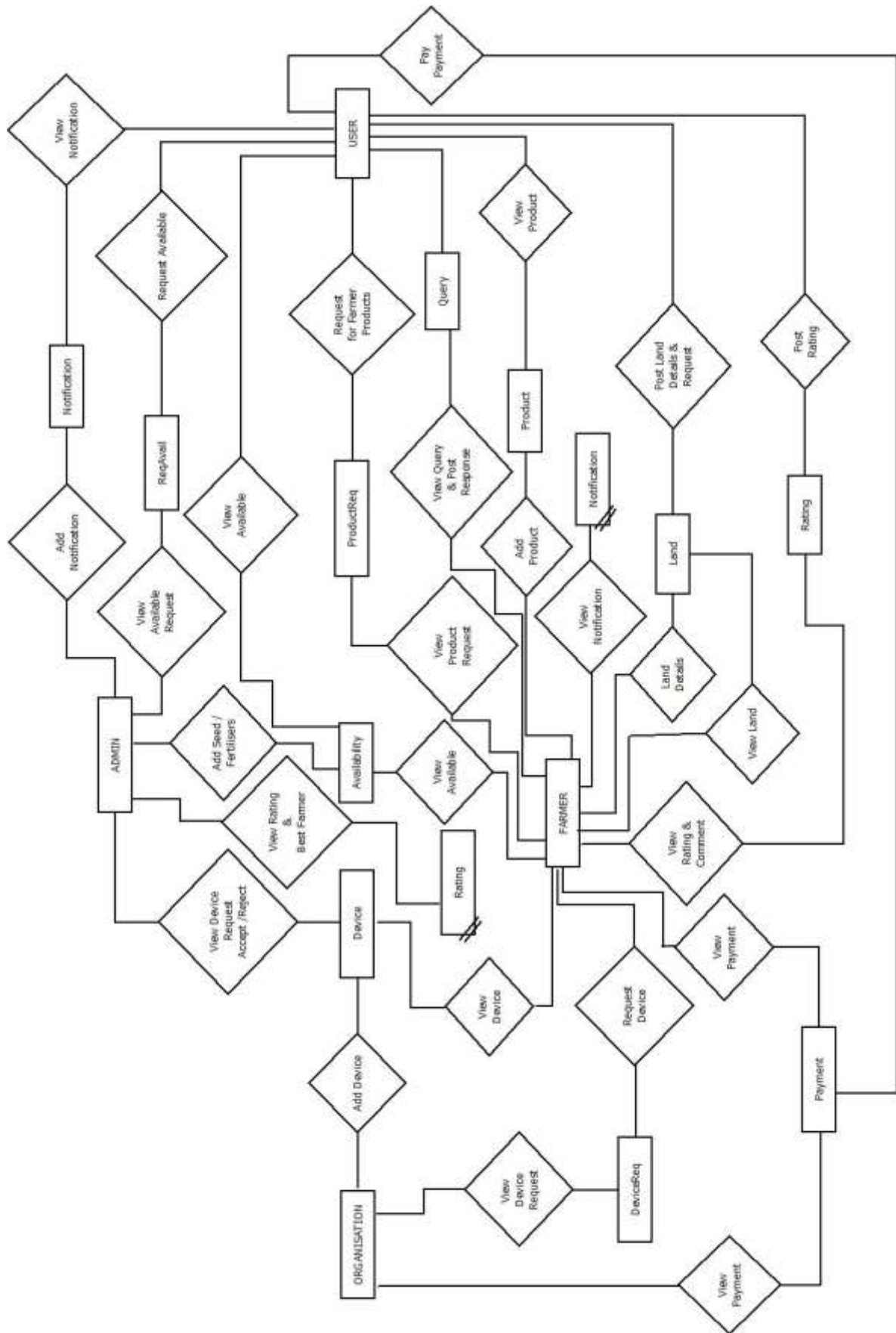


**ORGANISATION**

## ENTITY RELATIONSHIP MODEL

An entity-relationship model (ERM) is a theoretical and conceptual way of showing data relationships in software development. ERM is a database modeling technique that generates an abstract diagram or visual representation of a system's data that can be helpful in designing a relational database. These diagrams are known as entity-relationship diagrams, ER diagrams or ERDs.

An entity-relationship model (ERM) is a theoretical and conceptual way of showing data relationships in software development. ERM is a database modeling technique that generates an abstract diagram or visual representation of a system's data that can be helpful in designing a relational database. These diagrams are known as entity-relationship diagrams, ER diagrams or ERDs.



## DATABASE DESIGN

Database design is required to manage large bodies of information. The management of data involves both the definition of the structure of storage of information and provisions of mechanism for the manipulation of information. For developing an efficient database certain conditions have to be fulfilled such as:

- Control redundancy
- Ease of Use
- Data Independence
- Accuracy and Integrity

There are 5 major steps in design process

- Identify the table and relationship
- Identify the data that is needed for each and relationship
- Resolve the relationship
- Verify the design
- Implement the design
- The project uses the following database tables:

1. Login
2. Advert
3. Availability
4. Device
5. DeviceReq
6. Farmer
7. Land
8. Notification
9. Organisation
10. Payment
11. Product
12. ProductReq
13. Query
14. Rating
15. ReqAvail

## 16. User

The following tables lists the details of all database tables used in the project namely name of field, type, constraints used as well as brief description of the field.

**TABLE: 1**

Table name: Login

Primary key: LoID

Field name	Type	Size	Constraints	Description
LoID	int	10	Primary key	Unique id of the login table
loginId	int	10	null	Login ID
username	varchar	50	null	Username of the admin/student/staff
password	varchar	50	null	Password of the admin/student/staff
type	varchar	50	null	type

**TABLE: 2**

Table name: Advert

Primary key: AdID

Field name	Type	Size	Constraints	Description
AdID	int	10	Primary key	Unique id of the Advert table
itemID	int	10	null	Item ID
itemName	varchar	50	null	Item Name
itemDescription	varchar	50	null	Item Description
itemType	varchar	50	null	Item type
Price	int	10	null	Price of item

**TABLE: 3**

Table name: Availability

Primary key: AID

Field name	Type	Size	Constraints	Description
AID	int	10	Primary key	Unique id of the Availability table
productID	int	10	null	Product ID
productName	varchar	50	null	Product Name
productType	varchar	50	null	Product Type

**TABLE: 4**

Table name: Device

Primary key: DID

Field name	Type	Size	Constraints	Description
DID	int	10	primary key	Unique id of the Device table
deviceID	int	10	null	Device ID
deviceName	varchar	50	null	Device Name
description	varchar	50	null	Description of device

**TABLE: 5**

Table name: DeviceReq

Primary key: DRID

Field name	Type	Size	Constraints	Description
DRID	int	10	Primary key	Unique id of the DeviceReq table
deviceID	int	10	null	Device id
deviceName	varchar	50	null	Device name
description	varchar	50	null	Description of device
organisationID	varchar	50	null	Organisation ID
organisationName	varchar	50	null	Organisation Name

**TABLE: 6**

Table name: Farmer

Primary key: FID

Field name	Type	Size	Constraints	Description
FID	int	10	null	Unique id of the farmer table
farmerRegID	int	10	null	farmerReg id of the farmer table
Address	varchar	50	null	Address of farmer
dob	varchar	50	null	Dob of farmer
email	varchar	50	null	Email of farmer

contact	varchar	50	null	Contact of farmer
---------	---------	----	------	-------------------

**TABLE: 7**

Table name: Land

Primary key: LID

Field name	Type	Size	Constraints	Description
LID	int	10	primary key	Unique id of the study Land table
userID	int	10	null	User ID
farmerID	int	10	null	Farmer ID
address	varchar	50	null	Address of farmer/user
contact	varchar	50	null	Contact of farmer/user
email	varchar	50	null	Email of farmer/user
description	varchar	50	null	Description of land

**TABLE: 8**

Table name: Notification

Primary key: NID

Field name	Type	Size	Constraints	Description
NID	int	10	primary key	Unique id of the Notification table
notificationID	int	10	null	Notification id
addNotification	varchar	50	null	Add Notification
date	varchar	50	null	Date of notification



**TABLE: 9**

Table name: Organisation

Primary key: OID

Field name	Type	Size	Constraints	Description
OID	int	10	Primary key	Unique id of the Organisation table
organisationID	int	10	null	Organisation id
organisationName	varchar	50	null	Name of organisation
orgAddress	varchar	50	null	Address of organisation
contact	varchar	50	null	Contact of organisation
email	varchar	50	null	Email of organisation

**TABLE: 10**

Table name: Payment

Primary key: PayID

Field name	Type	Size	Constraints	Description
PayID	int	10	primary key	Unique id of the Payment table
userID	varchar	50	null	User ID
payamount	int	10	null	Payed Amount
date	varchar	50	null	Date of Payment

**TABLE: 11**

Table name: Product

Primary key: PID

Field name	Type	Size	Constraints	Description
PID	int	10	primary key	Unique id of the Product table
productID	int	50	null	Product ID
address	varchar	50	null	Address of user/farmer
dob	varchar	50	null	Dob of farmer/user
email	varchar	50	null	Email of farmer/user

**TABLE: 12**

Table name: ProductReq

Primary key: PRID

Field name	Type	Size	Constraints	Description
PRID	int	10	primary key	Unique id of the ProductReq table
productID	int	10	null	Id of product
productName	varchar	50	null	Name of Product

**TABLE: 13**

Table name: Query

Primary key: QID

Field name	Type	Size	Constraints	Description
QID	int	10	primary key	Unique id of the Query table
productID	int	10	null	Id of product
farmerID	int	10	null	Id of Farmer
farmerName	vchar	50	null	Name of farmer
query	vchar	50	null	Query from user

**TABLE: 14**

Table name: Rating

Primary key: RID

Field name	Type	Size	Constraints	Description
RID	int	10	primary key	Unique id of the Rating table
productID	int	10	null	Id of product
productName	vchar	50	null	Name of Product

farmerID	int	10	null	Id of Farmer
farmerName	varchar	50	null	Name of farmer
comment	varchar	50	null	Comments by User

**TABLE: 15**

Table name: ReqAvail

Primary key: ReID

Field name	Type	Size	Constraints	Description
ReID	int	10	primary key	Unique id of the ReqAvail table
productID	int	10	null	Id of product
productType	varchar	50	null	Type of product
productName	varchar	50	null	Name of product
description	varchar	50	null	Description of product

**TABLE: 16**

Table name: User

Primary key: UID

Field name	Type	Size	Constraints	Description
UID	int	10	primary key	Unique id of the User table
userID	int	10	null	Id of user
username	varchar	50	null	Username of user
address	varchar	50	null	Address of user
dob	varchar	50	null	Dob of user
contact	varchar	50	null	Contact of user
email	varchar	50	null	Email of user