A REPORT ON

**KRUSHI MITRA**

**Submitted in partial fulfillment of requirement**

**For the award of the degree of**

**Diploma Computer Engineering,**

**Gujarat Technological University**

**Submitted by:-**

**r**

***HARSHIL J. MAHAVADIA***

***(ENROLL NO: 156170307037)***

***VISHWESH CHOTALIYA***

***(ENROLL NO: 156170307018)***

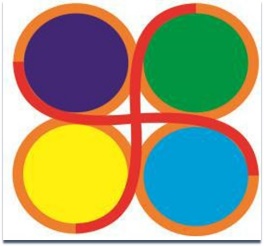
***KARTIK G. LUHAR***

***(ENROLLNO: 156170307036)***

**Under the guidance of**

**Internal Guide HOD**

Mr. Alpesh Thaker Mr. H.R. Parmar

****

**Computer Department**

**Government Polytechnic, Ahmedabad**

**November - 2017**

CERTIFICATE

This is to certify that the project entitled

***KRUSHI MITRA***

Submitted in partial fulfillment for the requirement of the degree of

**Diploma Computer Engineering**

Is a result of the bona fide work carried out by

**r**

***HARSHIL J. MAHAVADIA***

***(ENROLL NO: 156170307037)***

***VISHWESH CHOTALIYA***

***(ENROLL NO: 156170307018)***

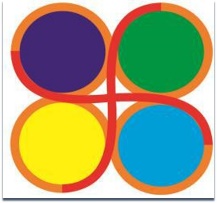
***KARTIK G. LUHAR***

***(ENROLLNO: 156170307036)***

During the academic session June to October 2014. They have undergone the process of shodh yatra, literature survey, problem definition and project designing. They are supposed to carry out the residue UDP Part-I work on same problem during Semester-V for the final fulfillment of the UDP work.

**Internal Guide HOD**

Mr. Alpesh Thaker Mr. H.R. Parmar

****

**Computer Department**

**Government Polytechnic, Ahmedabad**

**November – 2017**

ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude to Mr. Alpesh Thaker our internal guide for his help, support and guidance throughout this project. This project work would not have been successful without his valuable inputs. He was our source of inspirations during all phases of our project work.

We extend our heartfelt thanks to Mr. H.S Parmar, Head of xComputer Department for his co-operation in our project work.

We cannot forget to express our thanks to the entire staff of our department and colleagues for providing us a helping arm during the project work.

1. ***HARSHIL J. MAHAVADIA***
2. ***VISHWESH D. CHOTALIYA***
3. ***KARTIK G. LUHAR***

ABSTRACT

“Krushi Mitra” is a fully agricultural based system in which different users like farmer, customer, pesticide vendor, milk man gets a platform to contact each other.

In this system farmers can sell their products like crop online along with that they can get in touch with government subsides, news, weather, soil info, crop info, base price of products in market and able to give his/her price as input which customers can see.

Vendors are also able to contact farmers and customers if needed.

Along with that he/she is also able to see news, weather, base price of product in markets and able to give his/her price as input which customers can see.

Customers are also able to contact farmers, vendors and milk man if needed.

Along with that he/she is also able to see news, weather, and base price of product in markets and that of farmers or vendors.

This agricultural system is provided in two different languages: -

*1. English*

*2. Gujarati*

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| Chapter No. | Title |
| 1. | **Introduction** |
|  | Project Profile |
|  | Hardware Requirement |
|  | Software Requirement |
| 2. | **Literature Survey** |
|  | Existing System |
|  | Working of current system |
|  | Shortcomings of current system |
|  | Existing Site Survey |
|  | Process Model |
| 3. | **Proposed Website** |
|  | Introduction |
|  | Functionalities |
|  | Advantages |
|  | System Modules |
| 4. | **System Design** |
|  | System Flow Diagram |
|  | Entity Relationship Diagram(ERD) |
|  | Data Flow Diagram(DFD) |
|  | Use Case Diagram |
|  | Data Dictionary |
| 5. | **Conclusion and Future Scope** |
|  | Limitations of our Project |
|  | Conclusion |
|  | Future Scope |
|  | References and Bibliography |

Introduction

# PROJECT PROFILE

|  |  |
| --- | --- |
| **PROJECT TITLE:** | KRUSHI MITRA |
| **FRONT END:** | ANDROID(5.0 & ABOVE) |
| **BACK END:** | JAVA WEB SERVICES |
| **INTERNAL GUIDE:** | MR. ALPESH THAKER |
| **EXTERNAL GUIDE:** | **------** |
| **PROJECT DURATION:** | 1 YEAR |
| **TEAM SIZE:** | 3 MEMBERS |

# HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

* Processor: Intel CORE i3 or Higher version
* RAM: 4 GB or Higher
* Disk Space: 250 GB (Minimum)

SOFTWARE REQUIREMENTS:

* Operating System:- Android 5.0 & above

Literature survey

**EXISTING SYSTEM**

There is lack of this type of systems in India and have many limitations and drawbacks. Farmers are not getting proper knowledge of benefits provided by government and proper knowledge of farming. Thus, they are facing many problems

**WORKING OF CURRENT SYSTEM**

Many members of the Farmer department are involved in manual record keeping of the farmers. All the activities of the department are distributed among the staff members of the department. These activities include give advice, mange there record, show proper method of farming which is beneficial and profitable for farmer etc.

Existing system does not provide option to contact farmer, vendor, and customer with each other.

During the time of disasters Farmer department is not able to guide farmers and help them. Thus, farmer is not getting proper help or advice

In case of change of some farmers’ data, he/she must inform the farmer department about it. Then it takes actions to alter the old records.

Government must approach the students to pass any information and farmer have to physical come to Farmer department (Krushi Vibhag) to get any kind of information.

There are some advantages of manual system as described below:

ADVANTAGES:

* Less expensive to set up.
* The risk of corrupted data is much less.
* Data loss is less of a risk, particularly if records are stored in a fire-proof and protected environment.
* The process is simplified as you don't need to be familiar with computer operation.

SHORTCOMINGS OF CURRENT SYSTEM:

There are many drawbacks of this manual system also.

.

* Farmer physically go to farmer department for any query
* Paper records are difficult to manage and track.
* The physical exertion required to retrieve, alter, and re-file the paper records.
* It is almost difficult to back up all information.
* Information like farmer information, lab equipment, old notices, govt subsidies, advice portion etc. is not easily accessible to students.
* The current system does not provide accurate information. Sometimes due to the mistakes of the employees the data is lost.
* The process cannot be centralized, so it becomes very time consuming and cumbersome analysis.
* One of the major drawbacks of the current system is that it does not support online analysis of criticality data.
* Problems with duplicate copies of the same records are generally avoided.

**NEED FOR THE NEW SYSTEM**

As we have seen in the previous section, there are many drawbacks in the current manual system. So, there is a need to develop a website where most of the information can be easily available through a secure interface.

The purpose of “Krushi mitra” is to provide single window information and help centre to farmer customer and vendors. It enables user to compare market rates get in touch with news, weather and Govt subsidies.

It basically fulfils all the basic needs of farmer vendor and milkman including Govt policies, care centre, news, and live weather

It is available in both language English and Guajarati.

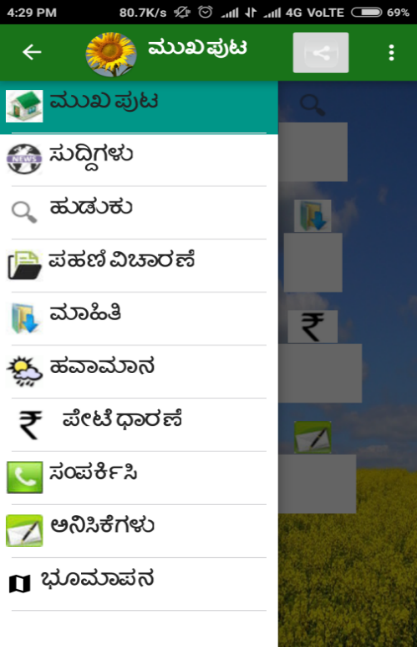
**EXISTING SITE SURVEY**

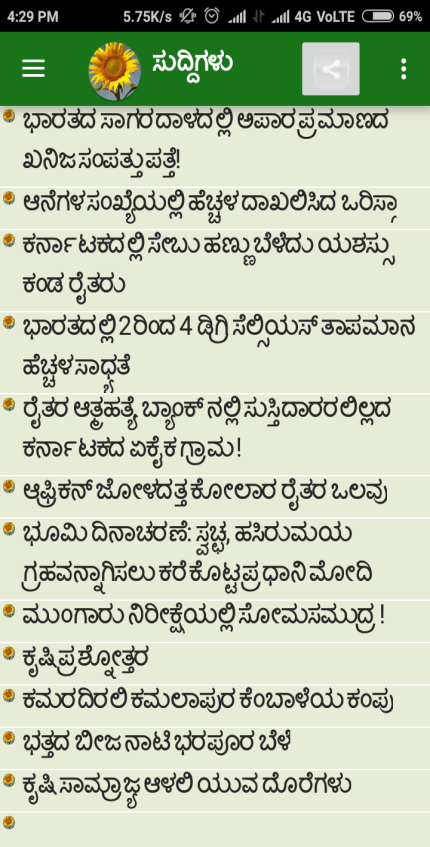
In the process of developing this website, we have studied following applications. We have searched the sites for how they help farmers and different user related to agricultural. We have taken many ideas from these sites and we have tried our best to incorporate many of these features in our applications, so our website can handle a department efficiently.

# Existing Site Survey

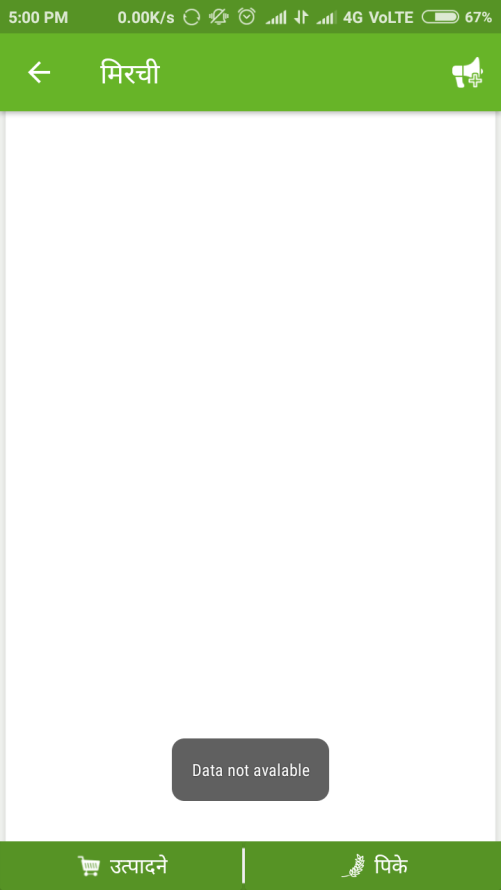
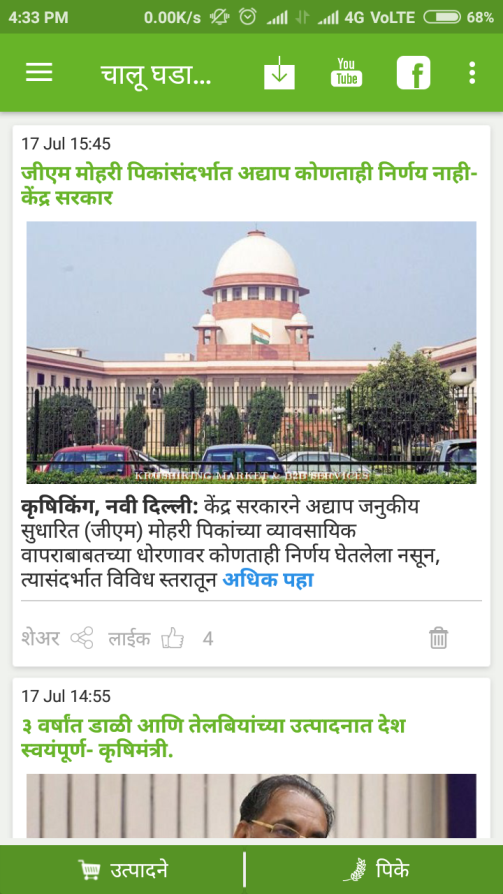
There are many applications all over world. We have studied following applications and we have tried to take good features from different existing applications and to incorporate in our system. Applications are developed for the whole university or a college. We have studied the departmental information from these applications.

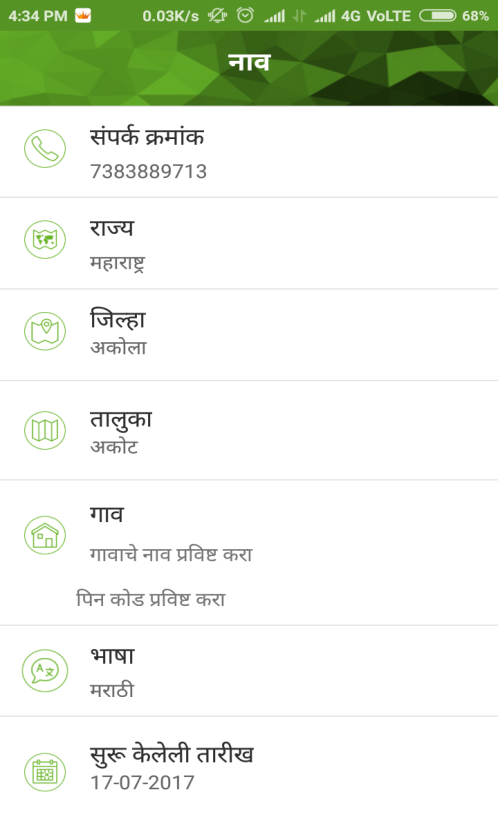
1. KRISHI MITRA:

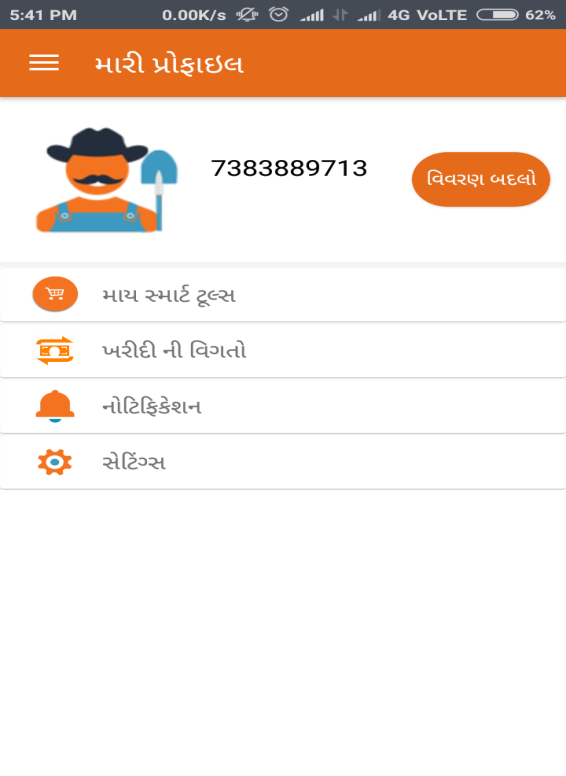


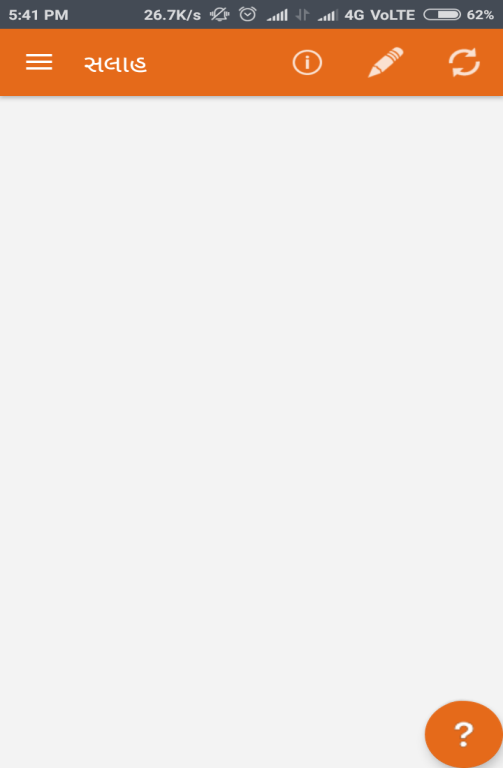
1. KRISHI KING:

1. RML FARMERS:



**PROCESS MODEL**

WHAT IS PROCESS MODEL?

A software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods, and tools layer to solve actual problems in an industry setting. A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required.

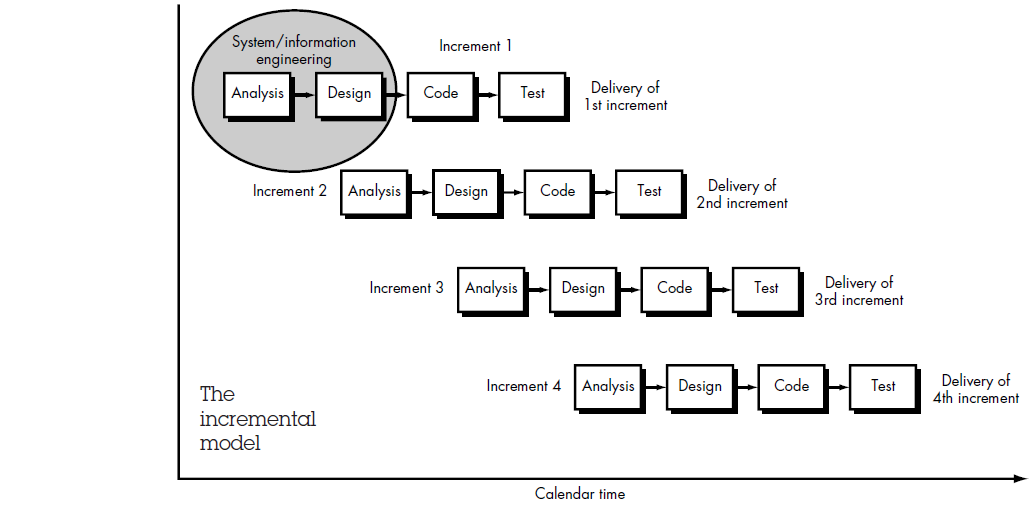
INCREMENTAL PROCESS MODEL:

The incremental model combines elements of the linear sequential model (applied repetitively) with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces a deliverable “increment” of the software.

When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features (some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed review). As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.

The incremental process model, like prototyping and other evolutionary approaches, is iterative in nature. But unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user.

DIAGRAM OF INCREMENTAL MODEL:



ADVANTAGES OF INCREMENTAL MODEL:

* Generates working software quickly and early during the software life cycle.
* This model is more flexible – less costly to change scope and requirements.
* It is easier to test and debug during a smaller iteration.
* In this model customer can respond to each built.
* Lowers initial delivery cost.
* Easier to manage risk because risky pieces are identified and handled during it’d iteration.

DISADVANTAGES OF INCREMENTAL MODEL:

* Needs good planning and design.
* Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
* Total cost is higher than waterfall.

WHEN TO USE THE INCREMENTAL MODEL:

* This model can be used when the requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some details can evolve with time.
* There is a need to get a product to the market early.
* A new technology is being used
* Resources with needed skill set are not available
* There are some high risk features and goals.

PROPOSED WEBSITE

**PROPOSED WEBSITE**

INTRODUCTION

The purpose of “Krushi Mitra” to replace the current paper records. Users are able to directly access all the services through a secure, online system.

The main objective of this website is to provide the online interface for farmer, customer, vendor, and milkman. It provides online buying and selling option and products and gives them a platform to contact each other.

There are options of Gujarati as well as English language. So, that if any users is English illiterate he can get the information in Gujarati.

The main Aim behind this is that the people in rural areas are far away from internet technology. So, to get all the information about agriculture collectively, we have developed an Application which will help the farmer, customer, vendor and milkman in many ways.

FUNCTIONALITIES:

The system utilizes user authentication, displaying only information necessary for an individual’s duties. Additionally, each sub-system has authentication allowing authorized users to create or update information in that subsystem.

All data is thoroughly reviewed and validated on the server before actual record alteration occurs.

Krushi Mitra application can be used as smart system which will be more sophisticated working for benefit of the users.

The Krushi Mitra application gives the whole information regarding to crops, soil, Weather status and also user can get the expert advice in Gujarati and in English languages.

The Krushi Mitra application provides a security platform in which farmer and vendors has to submit authorized document to prove that they are legal user.

Following are main functionalities available on the Application:

* Maintains records of Users
* Provides buying selling platform, weather, news, govt. subsidies, etc.
* Available in two languages

ADVANTAGES:

* This website can be used by users to perform different task like buying, selling, get advice, news, weather, govt. subsidies, soil/crop info, etc.
* The main objective of the software is storing and managing all the details of farmers as well as vendors, customers in a single system.
* It is developed for each person who is going to be affected by this system in agricultural field.
* This system integrates the existing manual or semi-automated process, starting from farmer's inquiry, registration, and allows them to contact their customers and get in touch with different services.

SYSTEM MODULES:

It maintains four levels of users: -

* Administrator Level
* Customer Level
* Vendor Level
* Farmer level

1. ADMIN MODULE:

Administrator is the main person who will manage this whole website.

This module is designed to help the administrator to manage the website efficiently.

Admin can manage all other types of users like Farmer, customer, vendor and it owns all the power to allow or disallow users for using website.

It has the responsibility to verify the data modified by any user before the actual change occurs in the database.

Admin can manage all other types of data like updating news, details of govt. subsidies, weather, base prices, and soil/crop info.

1. CUSTOMER LEVEL :

Customer can update his profile anytime.

He has power to contact farmers, vendors and any information like news, weather, advice, etc.

They can also view other user's details.

1. VENDOR LEVEL :

Vendor can update his profile anytime.

He has power to contact farmers, customer and any information like news, weather, advice, etc.

They can also view other user’s details. And is also able to insert his own price and able to see base price also

1. FARMER LEVEL :

Farmer can update his profile anytime.

He has power to contact vendor, customer and any information like news, weather, advice, etc.

They can also view other user’s details. And is also able to insert his own price and able to see base price also.

He is also able to see govt subsidies and soil/crop info.

SYSTEM DESIGN

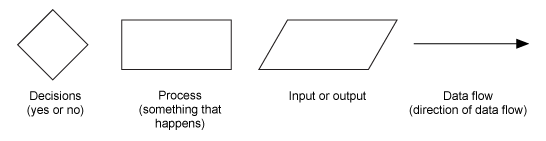
**SYSTEM FLOWCHART**

WHAT IS SYSTEM FLOWCHART?

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events.

System flowchart is the graphical representation of the flow of data in the system, and represents the work process of the system.

Various symbols are used in the flowchart to designate specific actions.



PARALLELOGRAM

Parallelogram is used to represent input and output of the system.

RECTANGLE

Rectangle represents the process that needs to be carried out in the system flowchart.

DIAMOND

Diamond indicates the decision to be performed in the flowchart.

OVAL

The oval shape signifies the start and end of the program in the system flowchart.

FLOW LINE

Flow line, a line with an arrowhead, is used to indicate the flow of data or logic in the system flowchart.

Following are the system flowchart of our Application:



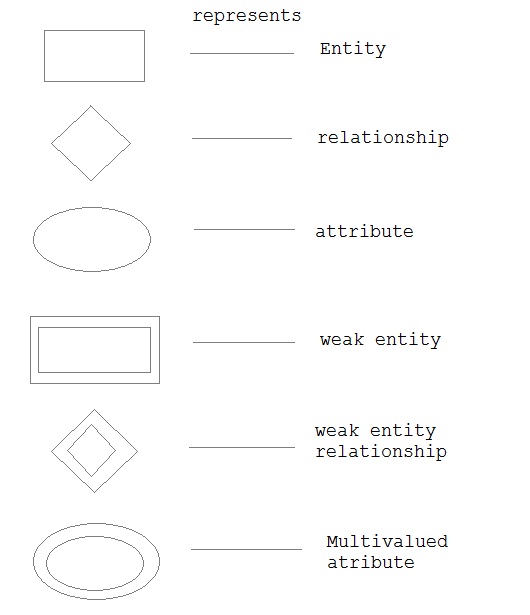


**E-R DIAGRAM**

WHAT IS E-R DIAGRAM?

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other.

ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, or ER diagrams or **ERDs**.



E-R SYMBOLS:

FOR EXAMPLE:

TWO RELATED ENTITIES

[https://upload.wikimedia.org/wikipedia/commons/thumb/3/3d/Erd-entity-relationship-example1.svg/283px-Erd-entity-relationship-example1.svg.png](https://en.wikipedia.org/wiki/File:Erd-entity-relationship-example1.svg)

AN ENTITY WITH AN ATTRIBUTE

[https://upload.wikimedia.org/wikipedia/commons/thumb/9/90/Erd-entity-with-attribute.svg/193px-Erd-entity-with-attribute.svg.png](https://en.wikipedia.org/wiki/File:Erd-entity-with-attribute.svg)

A RELATIONSHIP WITH AN ATTRIBUTE

[https://upload.wikimedia.org/wikipedia/commons/8/84/Erd-relationship-with-attribute.png](https://en.wikipedia.org/wiki/File:Erd-relationship-with-attribute.png)

[PRIMARY KEY](https://en.wikipedia.org/wiki/Primary_key)

[https://upload.wikimedia.org/wikipedia/commons/5/52/Erd-id-as-primary-key.png](https://en.wikipedia.org/wiki/File:Erd-id-as-primary-key.png)

Following are the ER diagram of our Application:

**DATA FLOW DIAGRAM**

WHAT IS DATA FLOW DIAGRAM (DFD)?

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its process aspects.

It is a graphical technique that depicts information flow and the transforms that are applied as data move form input to output.

DFD shows the interaction between the system and outside entities.

Data flow diagrams are commonly used during problem analysis.

It is also known as CONTEXT LEVEL DIAGRAM.

DFD SYMBOLS:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Example** |
| http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/entity.jpg | An **entity**. A source of data or a destination for data. | http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/entityeg.jpg |
| http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/process.jpg | A **process** or task that is performed by the system. | http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/processeg.jpg |
| http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/store.jpg | A **data store**, a place where data is held between processes. | http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/storeeg.jpg |
| http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/flow.jpg | A **data flow**. | http://cpanel.stpaulsscience.org/gceict/specifications/ocr/unit3/sdlc/dfd/floweg.jpg |

Following are the data flow diagram of our Application:

**DFD LEVEL 0**



**DFD LEVEL 1: FARMER**



**DFD LEVEL 1: CUSTOMER**



**DFD LEVEL 1: VENDOR**



**DFD LEVEL 1: ADMIN**



**USE CASE DIAGRAM**

WHAT IS USE-CASE DIAGRAM?

**Use case diagrams** are usually referred to as [behavior diagrams](https://www.uml-diagrams.org/uml-25-diagrams.html#behavior-diagram) used to describe a set of actions ([use cases](https://www.uml-diagrams.org/use-case.html)) that some system or systems ([subject](https://www.uml-diagrams.org/use-case-subject.html)) should or can perform in collaboration with one or more **external users** of the system ([actors](https://www.uml-diagrams.org/use-case-actor.html)). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

In a brief when we are planning to draw a use case diagram we should have the following items identified.

Functionalities to be represented as a use case Actors Relationships among the use cases and actors.

[USE-CASE SYMBOLS](http://creately.com/diagram-type/objects/use-case):

SYSTEM

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

****

USE CASE

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

****

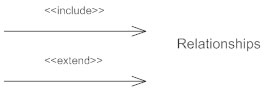
ACTORS

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

****

RELATIONSHIPS

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.

****

Following are the Use Case diagram of our Application:

**FARMER**

**CUSTOMER**



**VENDOR**



**ADMIN**



**DATA DICTIONARY**

WHAT IS DATA DICTIONARY?

Data dictionary describes all the data used in the system. A Data Dictionary is a document that describes the basic organization of a database.

Typically a data dictionary will contain a list of variables in the database as well as the assigned variable names and a description of each type of variable (e.g. character, numeric, dates).

The data dictionary should also include the values accepted for each variable, and any helpful comments such as important exclusions and skip patterns. The data dictionary is used primarily for data analysis.

Following table shows tables used in the system.

|  |  |
| --- | --- |
| Table Name | Description |
| Users | Stores the information of Users. |
| Roles | Stores the Multiple Roles of User. |
| Document | Stores the information of Documents of User. |
| Product\_master | Stores the information of Products. |
| User\_Products | Stores the information of Products of User. |
| Categories | Stores the Categories of Products. |
| Gov\_Subsidies | Stores the information of Government Subsidies. |
| Weather | Stores the information of Weather. |
| News | Stores the information of News. |
| Advice | Stores the information of Issues of User. |
| Authentication\_Token | Stores the information of Authentication Token. |
| Contact | Stores the information of Users. |
| Soil\_Info | Stores the information of Soil. |
| Crop\_Info | Stores the information of Crop. |
| Transaction\_Log | Stores the information of all Transaction. |

1. **Users**

**Primary key: User\_Id**

**Foreign key: Role\_Id and Doc\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **User\_Id** | **Int(8)** | **Primary Key** | **It stores Id of**  **Users** |
| 2. | Password | Varchar(50) | Not Null | Stores User's Password |
| 3. | Username | Varchar(50) | Not Null | Stores User's Name |
| 4. | Phone\_No | Varchar(15) | Not Null | Stores User's Contact No |
| 5. | Email\_Id | Varchar(50) | Not Null | Stores User's Email Id |
| 6. | City | Varchar(15) | Not Null | Stores User's city |
| 7. | Role\_Id | Int(8) | Foreign Key | Stores Id of User's Role |
| 8. | Doc\_Id | Int(8) | Foreign Key | Stores User's Document Id |
| 9. | Approval\_Status | Boolean | Not Null | Stores User's Document Approval Status |
| 10. | Approved\_By | Varchar(50) | Not Null | Stores Name of Admin |
| 11. | Approval\_Date\_Time | Datetime | Not Null | Stores User's Document Approval Date Time |
| 12. | Approval\_Remarks | Varchar(50) | Not Null | Stores Remarks of Document |
| 13. | Created\_At | Datetime | Not Null | Stores User Created Date Time |
| 14. | Updated\_At | Datetime | Not Null | Stores User Updated Date Time |

1. **Roles**

**Primary key: Role\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Role\_Id** | **Int(8)** | **Primary Key** | **It stores Id of User's Role** |
| 2. | Role\_Name | Varchar(50) | Not Null | Stores User's Name |
| 3. | Created\_At | Datetime | Not Null | Stores User's Role Created Date Time |
| 4. | Updated\_At | Datetime | Not Null | Stores User's Role Updated Date Time |

1. **Document**

**Primary key: Doc\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Doc\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of User's Document** |
| 2. | Doc\_Name | Varchar(50) | Not Null | Stores User's Document Name |
| 3. | Description | Varchar(MAX) | Not Null | Stores User's Document Description |
| 4. | Created\_At | Datetime | Not Null | Stores User's Document Created Date Time |
| 5. | Updated\_At | Datetime | Not Null | Stores User's Document Updated Date Time |

1. **Product\_Master**

**Primary key: Product\_Id**

**Foreign key: Category\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Product\_Id** | **Int(8)** | **Primary Key** | **It stores Id of Product** |
| 2. | Product\_Name | Varchar(50) | Not Null | Stores Name of Product |
| 3. | Category\_Id | Int(8) | **Foreign Key** | Stores Category Id |
| 4. | Product\_Image | Blob | Not Null | Stores Image of Product |
| 5. | Base\_Price\_Per\_Unit | Varchar(20) | Not Null | Stores Base Price Per Unit |
| 6. | Basic\_Unit | Varchar(20) | Not Null | Stores Basic Unit |
| 7. | Created\_At | Datetime | Not Null | StoresProduct Base Price Created Date Time |
| 8. | Updated\_At | Datetime | Not Null | StoresProduct Base Price Updated Date Time |

1. **User\_Products**

**Primary key: User\_Product\_Id**

**Foreign key: Product\_Id and User\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **User\_Product\_Id** | **Int(8)** | **Primary Key** | **It stores Id of User Products** |
| 2. | Product\_Id | Int(8) | **Foreign Key** | Stores Product Id |
| 3. | User\_Id | Int(8) | **Foreign Key** | Stores User Id |
| 4. | Description | Varchar(MAX) | Not Null | Stores Product Description |
| 5. | Product\_Price\_Per\_Unit | Varchar(20) | Not Null | Stores Product Price Per Unit |
| 6. | Created\_At | Datetime | Not Null | Stores Product Price Created Date Time |
| 7. | Updated\_At | Datetime | Not Null | Stores Product Price Updated Date Time |

1. **Categories**

**Primary key: Category\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Category\_Id** | **Int(8)** | **Primary Key** | **It stores Id of Category** |
| 2. | Category\_Name | Varchar(50) | Not Null | Stores Category Name |
| 3. | Created\_At | Datetime | Not Null | Stores Category of Product Created Date Time |
| 4. | Updated\_At | Datetime | Not Null | Stores Category of Product Updated Date Time |

1. **Gov\_Subsidies**

**Primary key: Gov\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Gov\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of Government Subsidies** |
| 2. | Title | Varchar(20) | Not Null | Stores Government Subsidies Title |
| 3. | Description | Varchar(MAX) | Not Null | Stores Government  Subsidies Description |
| 4. | Created\_At | Datetime | Not Null | StoresGovernmentSubsidies Created Date Time |
| 5. | Updated\_At | Datetime | Not Null | StoresGovernmentSubsidies Updated Date Time |

1. **Weather**

**Primary key: Weather\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Weather\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of Weather** |
| 2. | Title | Varchar(20) | Not Null | Stores Government Subsidies Title |
| 3. | Description | Varchar(MAX) | Not Null | Stores Weather  Subsidies Description |
| 4. | Created\_At | Datetime | Not Null | Stores Weather Created Date Time |
| 5. | Updated\_At | Datetime | Not Null | Stores Weather Updated Date Time |

1. **News**

**Primary key: News\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **News\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of News** |
| 2. | Title | Varchar(20) | Not Null | Stores News Title |
| 3. | Description | Varchar(MAX) | Not Null | Stores NewsDescription |
| 4. | Created\_At | Datetime | Not Null | Stores News Created Date Time |
| 5. | Updated\_At | Datetime | Not Null | Stores News Updated Date time |

1. **Advice**

**Primary key: Advice\_Id**

**Foreign key: User\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Advice\_Id** | **Int(8)** | **Primary Key** | **It Stores Id Advice** |
| 2. | User\_Id | Int(8) | **Foreign Key** | Stores Id of User |
| 3. | Issue | Varchar(20) | Not Null | Stores Advice Issue |
| 4. | Description | Varchar(MAX) | Not Null | Stores Advice Description |
| 5. | Created\_At | Datetime | Not Null | Stores Issue Created Date Time |
| 6. | Updated\_At | Datetime | Not Null | Stores Issue Updated Date Time |

1. **Authentication\_Token**

**Primary key: Auth\_Token\_Id**

**Foreign key: User\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Auth\_Token\_Id** | **Int(8)** | **Primary Key** | **It stores Id of Authentication Token** |
| 2. | Auth\_Token | Varchar(50) | Not Null | Stores Authentication Token Name |
| 3. | User\_Id | Int(8) | **Foreign Key** | Stores User Id |
| 4. | Device\_Id | Varchar(500) | Not Null | Stores Device Id |
| 5. | Device\_Type | Varchar(50) | Not Null | Stores device name |
| 5. | Created\_At | Datetime | Not Null | Stores Token Created Date |
| 6. | Updated\_At | Datetime | Not Null | Stores Token Updated Date |

1. **Contact**

**Primary key: Unique\_Id**

**Foreign key: User\_Id and Contact\_Person\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Unique\_Id** | **Int(8)** | **Primary Key** | **It Stores Unique Id** |
| 2. | User\_Id | Int(8) | **Foreign Key** | Stores User Id |
| 3. | Contact\_Person\_  Id | Int(8) | **Foreign Key** | Stores Contact\_person Id |
| 4. | Date\_Time | Datetime | Not Null | Stores Contact Date Time |
| 4. | Description | Varchar(MAX) | Not Null | Stores User's Contact Description |
| 5. | Created\_At | Datetime | Not Null | Stores Contact Created Date Time |
| 6. | Updated\_At | Datetime | Not Null | Stores Contact Updated Date Time |

1. **Soil\_Info**

**Primary key: Soil\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Soil\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of Soil** |
| 2. | Soil\_Name | Varchar(50) | Not Null | Stores Soil Name |
| 3. | Soil\_Image | Blob | Not Null | Stores Soil Image |
| 4. | Description | Varchar(MAX) | Not Null | Stores User's Soil Description |
| 5. | Created\_At | Datetime | Not Null | Stores Soil Information Created Date Time |
| 6. | Updated\_At | Datetime | Not Null | Stores Soil Info. Updated Date Time |

1. **Crop\_Info**

**Primary key: Crop\_Id**

**Foreign key: ---**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Crop\_Id** | **Int(8)** | **Primary Key** | **It Stores Id of Crop** |
| 2. | Crop\_Name | Varchar(20) | Not Null | Stores Crop Name |
| 3. | Crop\_Image | Blob | Not Null | Stores Crop Image |
| 4. | Description | Varchar(MAX) | Not Null | Stores Crop Description |
| 5. | Created\_At | Datetime | Not Null | Stores Crop Information Created Date Time |
| 6. | Updated\_At | Datetime | Not Null | Stores Crop Information Updated Date Time |

1. **Transaction\_Log**

**Primary key: Transaction\_Id**

**Foreign key: User\_Id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Column Name | Type | Constraint | Description |
| 1. | **Transaction\_Id** | **Int(8)** | **Primary Key** | **It stores Id of Transaction** |
| 2. | User\_Id | Int(8) | **Foreign Key** | Stores User Id |
| 3. | Date\_Time | Datetime | Not Null | Stores Transaction Date Time |
| 4. | Transaction\_Type | Varchar(20) | Not Null | Stores Transaction Type |
| 5. | Transaction\_Query | Varchar(500) | Not Null | Stores Transaction Query |
| 6. | Previous\_Value | Varchar(20) | Not Null | Stores Transaction Previous Value |
| 7. | Created\_At | Datetime | Not Null | Stores Transaction Created Date Time |
| 8. | Updated\_At | Datetime | Not Null | Stores Transaction Updated Date |

Conclusion and future scope

**LIMITATIONS**

* The proposed system is developed to overcome the shortcomings of the manual system, but still there are many limitations to this new system.
* The limitations are as follows:
* This website is developed for a Gujarat level only in two languages gujarati and English.
* It doesn’t have the features like payment option etc.
* Many features can be made more dynamic.

**FUTURE SCOPE**

* As we discussed in the limitations section, we can upgrade the website to incorporate those changes.
* We can make the website more dynamic and we can also extend it to manage more than one department.
* In future payment option is provided and option to contact Krushi Vibhag.
* In future the system will be provided in many different languages.
* We have also designed our data dictionary to store most the data but when we will

**CONCLUSION**

* implement the system in the next semester, it may be possible that data dictionary may change,
* We have tried our best to make our website user friendly and useful to manage a department.

**BIBLIOGRAPHY**

Web Reference (website names):

1. <http://www.google.com>
2. <http://www.W3schools.com>
3. [http://nevonprojects.com](http://nevonprojects.com/project-ideas/android-project-ideas/)
4. <http://www.lovelycoding.org>
5. <http://1000projects.org>

Books: