**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** |  |  | **Contents** |
| 1 |  |  | ACKNOWLEDGEMENT |
| 2 |  |  | INTRODUCTION |
|  | 2.1 |  | Synopsis |
|  | 2.2 |  | Software And Hardware Requirements |
| 3 |  |  | SCOPE AND OBJECTIVE |
| 4 |  |  | THEORITICAL BACKGROUND |
| 5 |  |  | PROBLEM DEFINITION |
|  | 5.1 |  | Drawbacks of Existing System |
|  | 5.2 |  | Advantages of Proposed System |
| 9 |  |  | SYSTEM ANALYSIS AND DESIGN |
|  | 9.2 |  | E-R Diagram |
|  | 9.3 |  | Class Diagram |
|  | 9.4 |  | Use Case Diagram |
|  | 9.5 |  | Sequence Diagram |
|  | 9.8 |  | Activity Diagram |
|  | 9.13 |  | Screen Design |
| 10 |  |  | PROCESS INVOLVED(Description of modules in detail) |
| 12 |  |  | FUTURE ENHANCEMENTS |
| 13 |  |  | REFERENCE AND BIBLIOGRAPHY |

**Introduction**

The project is a website which name is “E- agriculture”. As this is an ecommerce website it will play an important role for the user to shop online through this website . In this website a user will be enabled to shop efficiently because modules will be defined properly which will enhance the shopping .

Todays farmers are unpaid for the crops they grow. The investment done by the farmers in the crops and the ROI (Return of investment) which they receive from trader leads to big loss.

**MODULES**

• **Farmer’s inventory**

In this project, the system is designed in such a way that whenever there is a shortage of goods in the inventory, the farmer will be notified about it. This will help the farmer to refill the stocks accordingly.

• **Customer’s inventory**

Customer can choose from the list of products available which are uploaded by farmers. They can put the products in their cart to purchase it latter.

• **Product Details**

Farmer can update the product information of their uploaded products. They can increase the quantity or can increase or decrease the expiry date of the products. Farmers can increase or decrease the cost of the products according to the market conditions or according to their profit margins.

• **Customer’s Cart**

The products selected by the customers will first go in their cart before making purchase of any product. Customers can increase or decrease the quantity of the products before making any purchase from this website.

**EXISTING SYSTEM**

Farmer (many of them) take credit for almost a many things like seeds, inputs, fertilizers etc.

The crop is ready, and this is the time he has to pay his debt (He can't store his product as debts to pay and no place for storage).

He loads the crop in a tractor or any vehicle and go to mandi (Keep in mind the produce is in very large quantities as in quintals and tonnes)

The transportation is not free (So no option of taking the produce back home) The market is flooded with the produce so practically all the warehouses are full (storage out of question).

And a person sitting in APMC mandi (agriculture produce marketing committee) quotes some random price which are not acceptable to the farmer then he has no choice.

He has to sell it at whatever price quoted by that agent (called commission agent CA).

Today’s farmers are unpaid for the crops they grow. The investment done by the farmers in the crops and the ROI (Return of investment) which they receive from trader leads to big loss.

Farmers sell 1KG potatoes at the price of 1 to 2 RS/KG to the traders then trader bears the transportation cost and adding his profit which leads to increase the price by 10% now the distributors will distribute the crops in local market will try to earn his profit and hence increase the cost of the crops by another 10%

The local market vendors of the crops will again increase the price per KG by another 10% and hence the customer receives the same potatoes at the rate of 15 to 20 RS 1KG.

Hence, the farmers do not get proper prices for their products.

**TECHNOLOGY USED**

**Front end:** Bootstrap (version 3.6).

**Back end:** PHP5.

**Database & Server:** MySQL database and XAMPP server.

**BOOTSTRAP** **v3.6 :** Bootstrap is a framework build on HTML5 and CSS3. It is used to make responsive UI for web development. In this project entire front-end is made using bootstrap.

**PHP5 :** It is used to communicatewith database. It makes easy for us to transfer data easily from front-end to back-end.

**MySql :** This is the database which is used in the project. MySql database is easy to pickup and easily accessible.

**XAMPP Server :** Xampp server is used so as to get the HTTP access to transfer data.

**CRUD** represents an acronym for the database operation Create, Read, Update and Delete.

Using this website the users can check the best goods and all the transaction history whereas farmers can get their profit.

There will be smooth online transaction for the users, records will be safely stored in database.

PROPOSED SYSTEM

As this is .an ecommerce website, it will play an important role for the user to shop online through this website. In this website a user will be enabled to shop efficiently because modules will be defined properly which will enhance the shopping.

The system proves to be best when all things comes on the ground transparency. There will be no chance of theft which could be done by the retailers or distributers. Bill of every transaction will be maintained and product details will be at the edge of products correctness. Products received at the end by the customers will be at a reasonable cost even-after the product includes all the cost including transportation, GST, retailers or distributers commission.

Ecommerce portal benefits to the farmers as they can set the price of the products as per their will, there will be no governing body on farmers to set the price of their products. Farmers will get notification of their products if at any time the quantity of their products come down less then 5 per kg.

Online system makes it possible for the farmers to earn the right profit without suffering any further loss when compared to the existing system.

**ABSTRACT**

The main objectives of this project are:

* Online shopping will help them to search crop at reasonable price
* Farmers will get direct profit without spending much
* Buying and selling between customer and farmers will be easy
* Customers satisfaction
* Helps to keep the history of all the goods from both the ends i.e. customer and farmers.
* The main objective of the project is to help the farmers achieve greater value for their products.
* To facilitate market interventions for value addition to farm produce.
* To encourage the intellectual development of the farmers.
* Orientation of input dealers on location-specific crop production technologies of broad-based agriculture with reference to field problems.
* To help farmers to achieve greater value for their products.

**SYSTEM DESIGN**

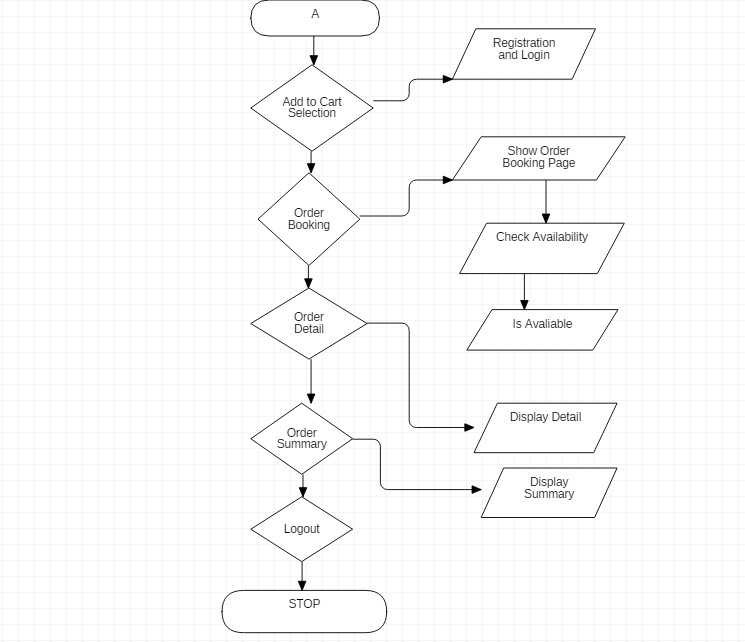
**FLOW CHART**

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams

Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

Below are some of the common flowchart symbols.

|  |  |  |  |
| --- | --- | --- | --- |
| **TERMINAL/TERMINATOR** |  | | |
| **PROCESS** |  | | |
|  |  |  |
| **DOCUMENT** |  | | |
| **DATA, OR INPUT/OUTPUT** |  | | |
| **FLOW ARROW** |  | | |



**USE CASE DIAGRAM**

Use Case diagrams show the various activities the users can perform on the system. The

System is something that performs a function. They model the dynamic aspects of the system. It provides a user’s perspective of the system. Use Case Diagrams model the functionality of system by using Actors and Use Cases. An actor may participate in more than one use case and, conversely, more than one actor may participate in the same use case. In short, Use Case Diagram is a graphical depiction with an accompanying textual description of use cases and the actor that participate in them.

* **Actor:** Actor is a user of the system. The actor may be a person or another external system. Actor is a role that a particular user plays while interacting with the system.
* **Use case:** Use Cases are services or functions provided by the system to its users.

Each use case describes one logical interaction between the Actor and the system.

* **Relationship:** Relationships are simply illustrated with a line connecting actors to use cases.

A Use Case diagram has the following elements:

|  |
| --- |
| \_ **Stick figure:** representing an Actor |
| \_ **Oval:** representing a use case |
| \_ **Association lines:** representing communication between Actors and Use Cases |

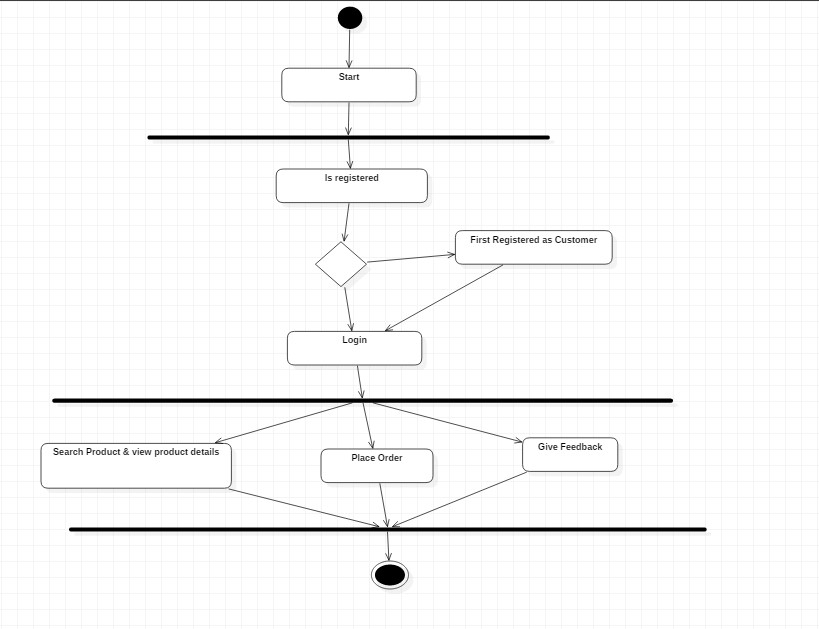
**Activity Diagram**

Activity diagrams are graphical representations of workflows of stepwise activities and action with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

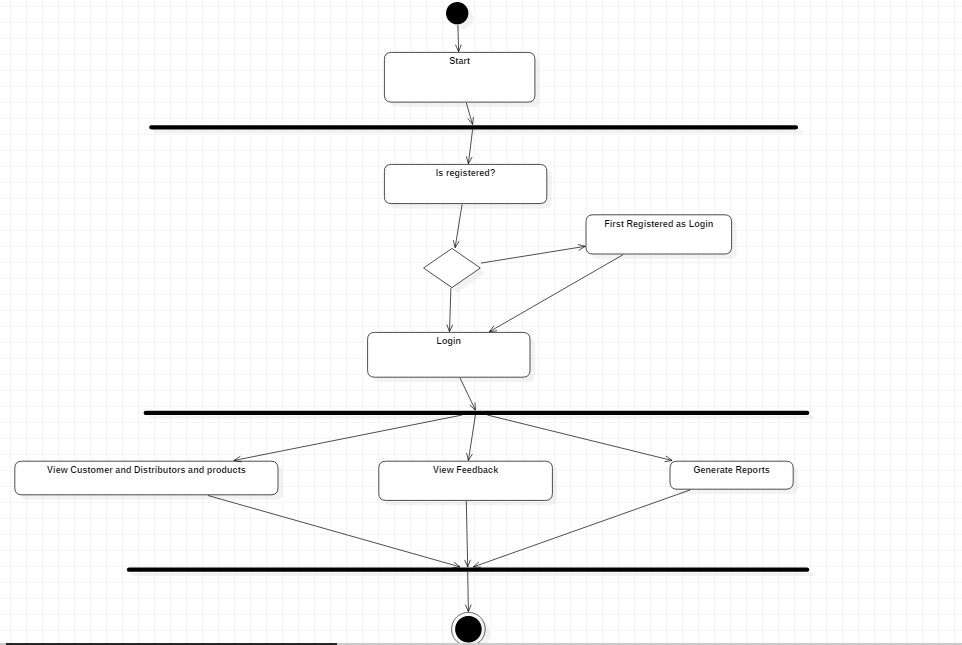
|  |
| --- |
| • Rounded rectangles represent actions |
| • Diamonds represent decisions |

|  |  |
| --- | --- |
| • | Bars represent the start (split) or end (join) of concurrent activities |
| • | A black circle represents the start (initial state) of the workflow |
| • | An encircled black circle represents the end (final state). |
|  | Arrows run from the start towards the end and represent the order in which activity happens. |

**CUSTOMER ACTIVITY DIAGRAM**



**ADMIN ACTIVITY DIAGRAM**:



**E – R DIAGRAM**

An Entity-Relationship diagram (E-R diagram) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ER diagram is a conceptual and representational model of data used to represent the entity framework infrastructure.

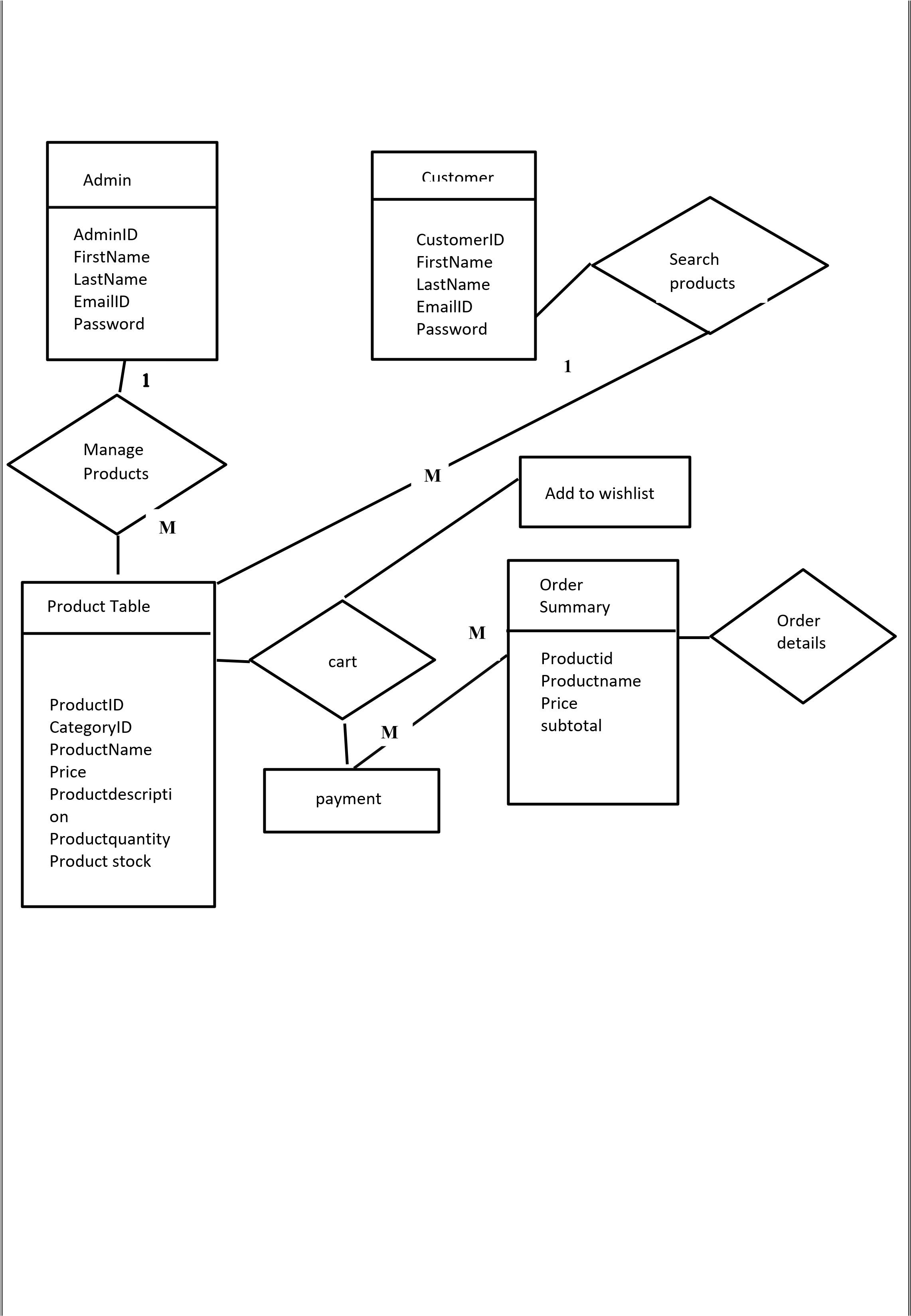
The elements of an E-R diagram are:

* Entities
* Relationships
* Attributes

An entity-relationship diagram is crucial to creating a good database design. It is used as a high-level logical data model, which is useful in developing a conceptual design for databases.

An entity is a real-world item or concept that exists on its own. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.

An attribute of an entity is a particular property that describes the entity. A relationship is the association that describes the interaction between entities. Cardinality, in the context of E-R diagram, is the number of instances of one entity that can, or must, be associated with each instance of another entity. In general, there may be one-to-one, one-to-many, or many-tomany relationships.



Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

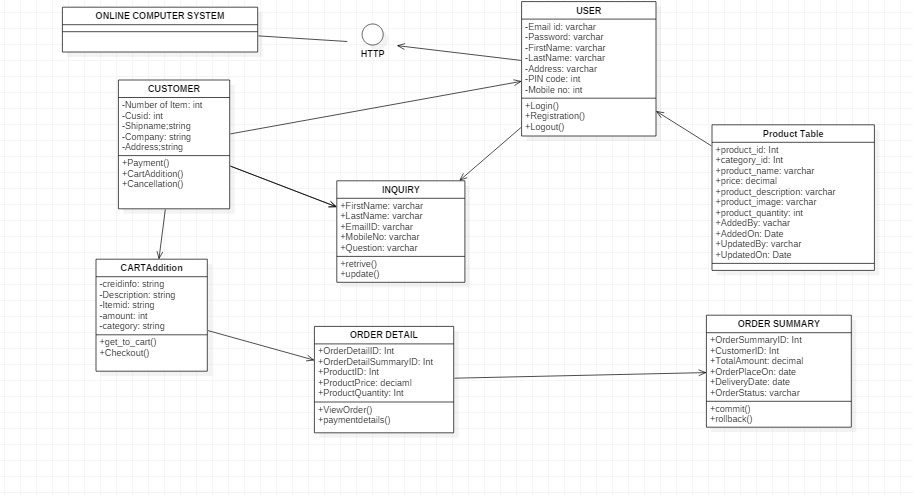
In the diagram, classes are represented with boxes that contain three compartments:

* The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

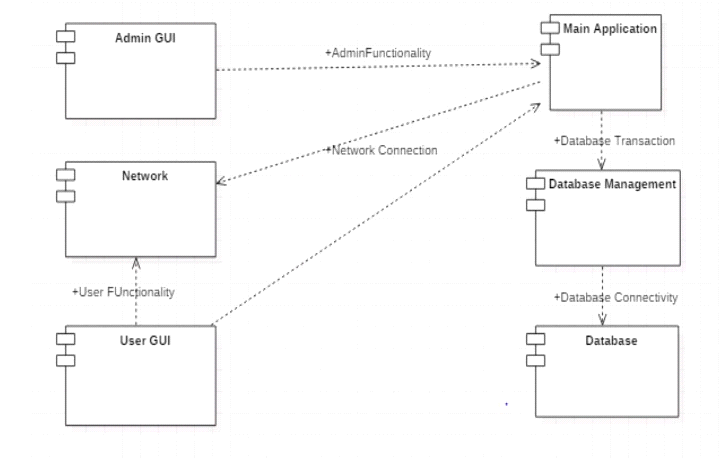
* The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

**CLASS DIAGRAM**



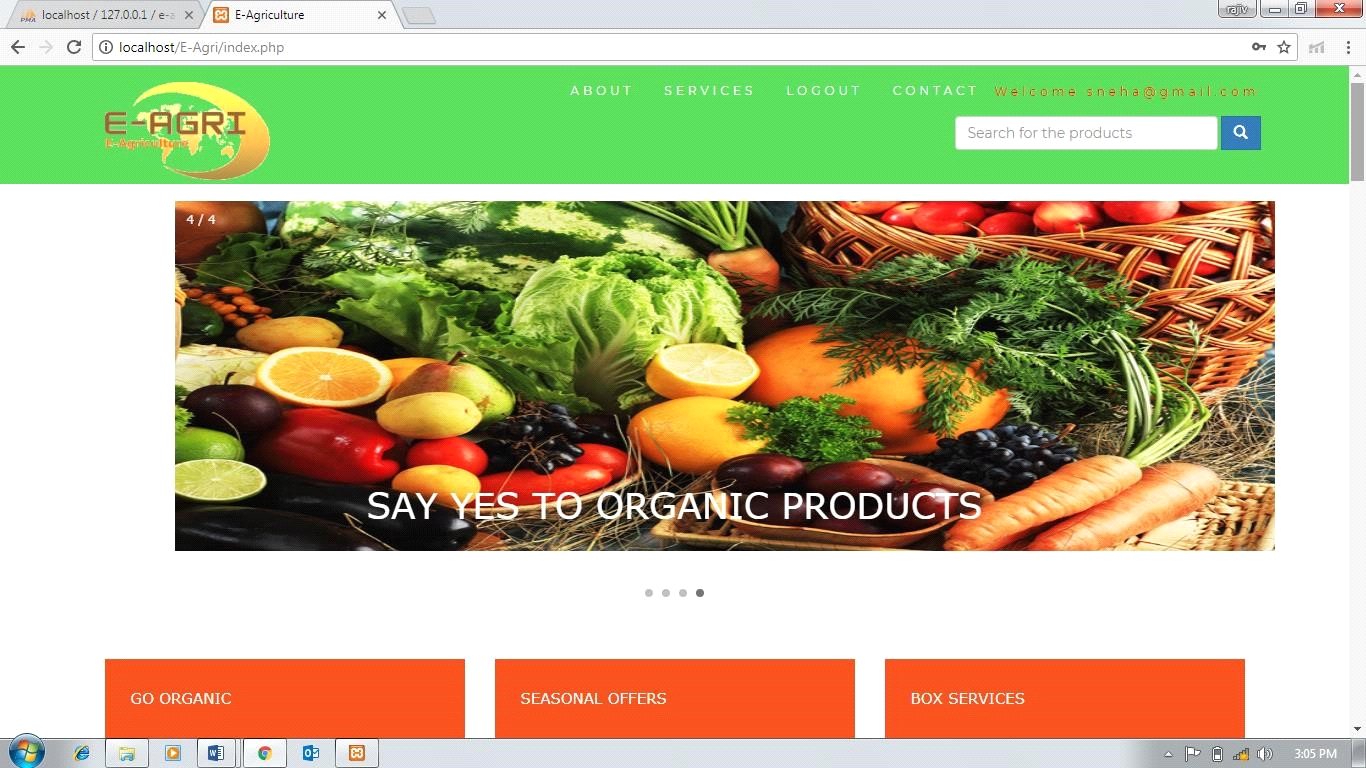
**Component Diagram**

* Describe organization and dependencies among physical software components, such as source code, run-time (binary) code, and executables.

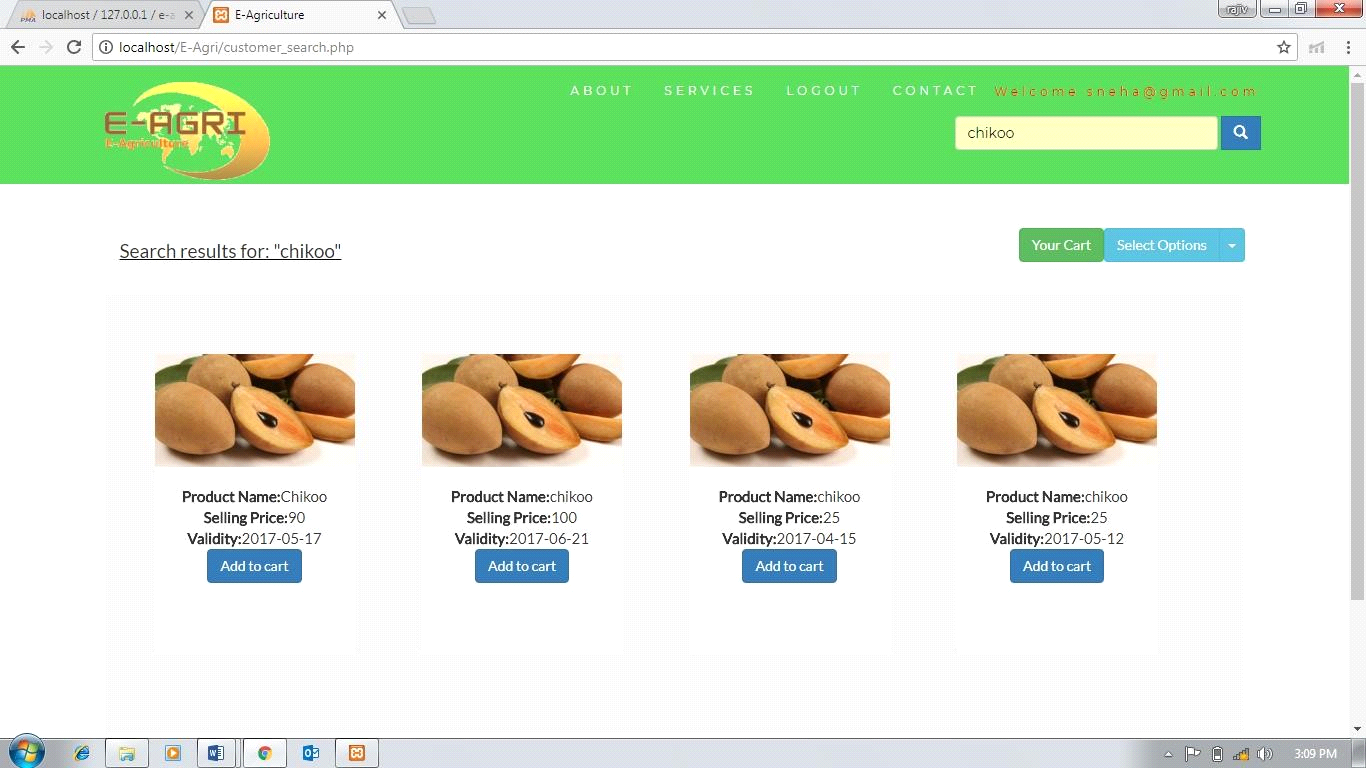


**SCREENSHOTS**

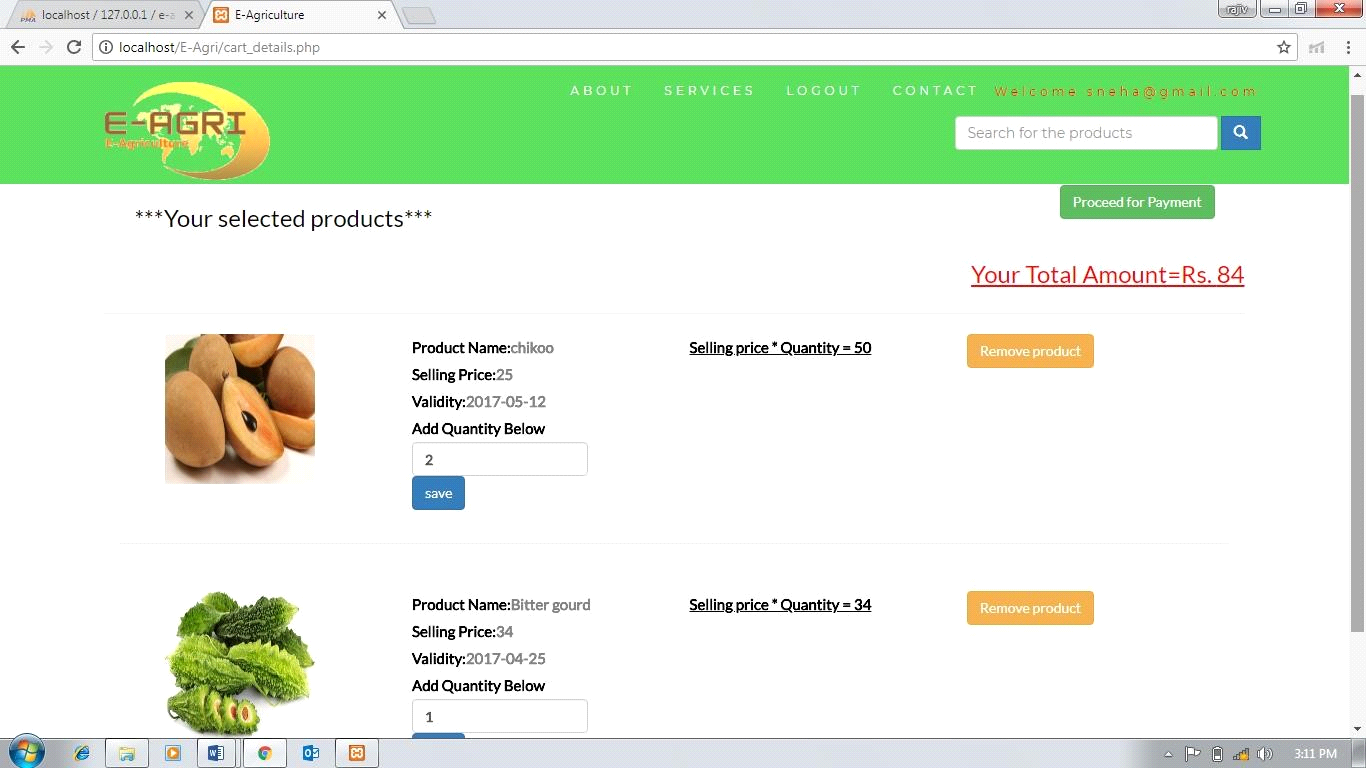
**First Page After Login**



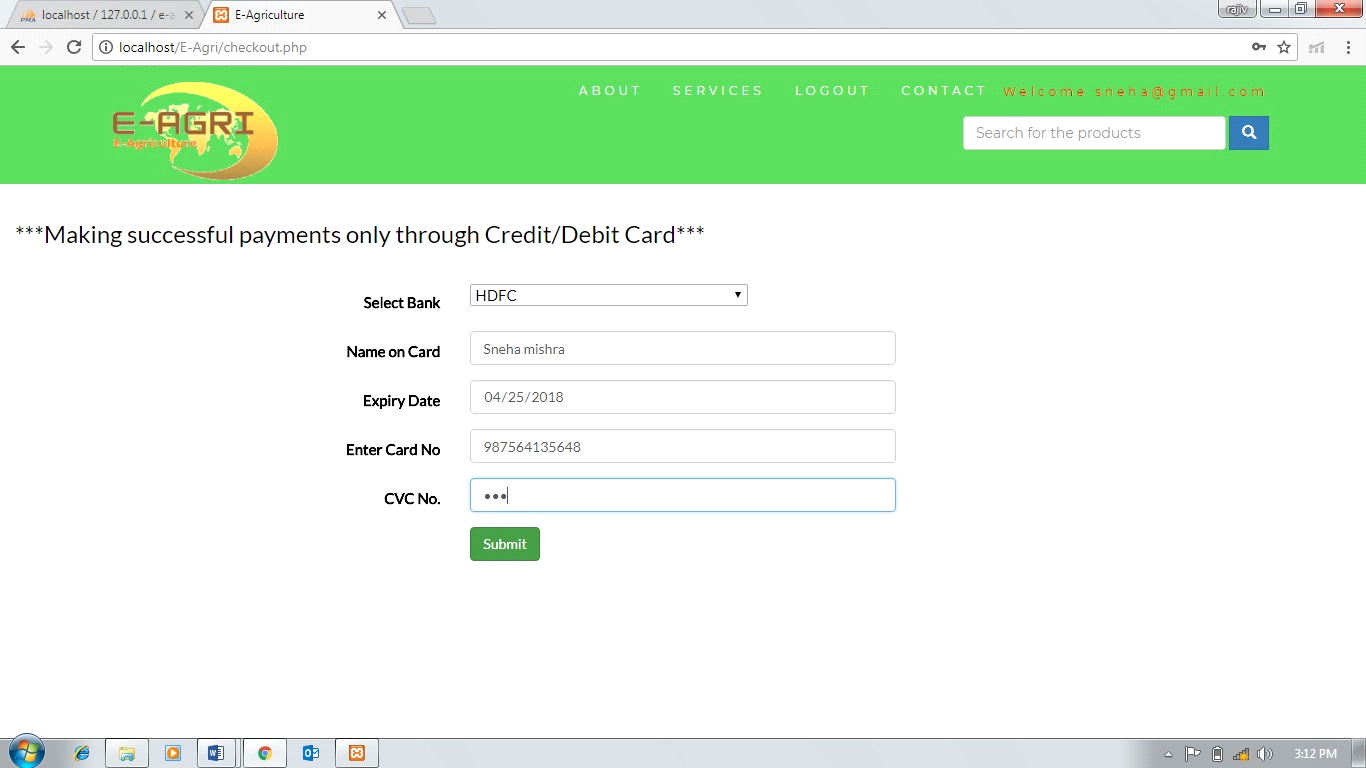
**Search Result Page For Product**



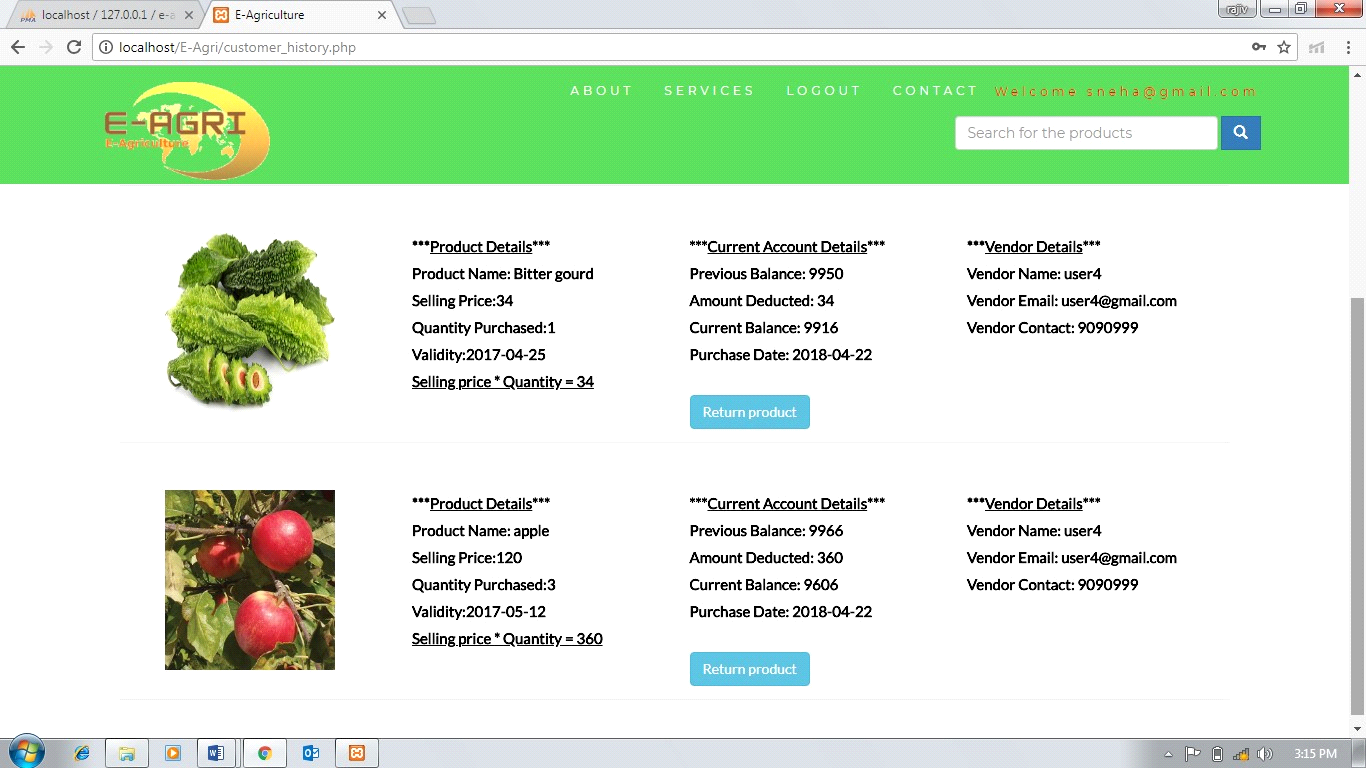
**Customer’s Cart Page**



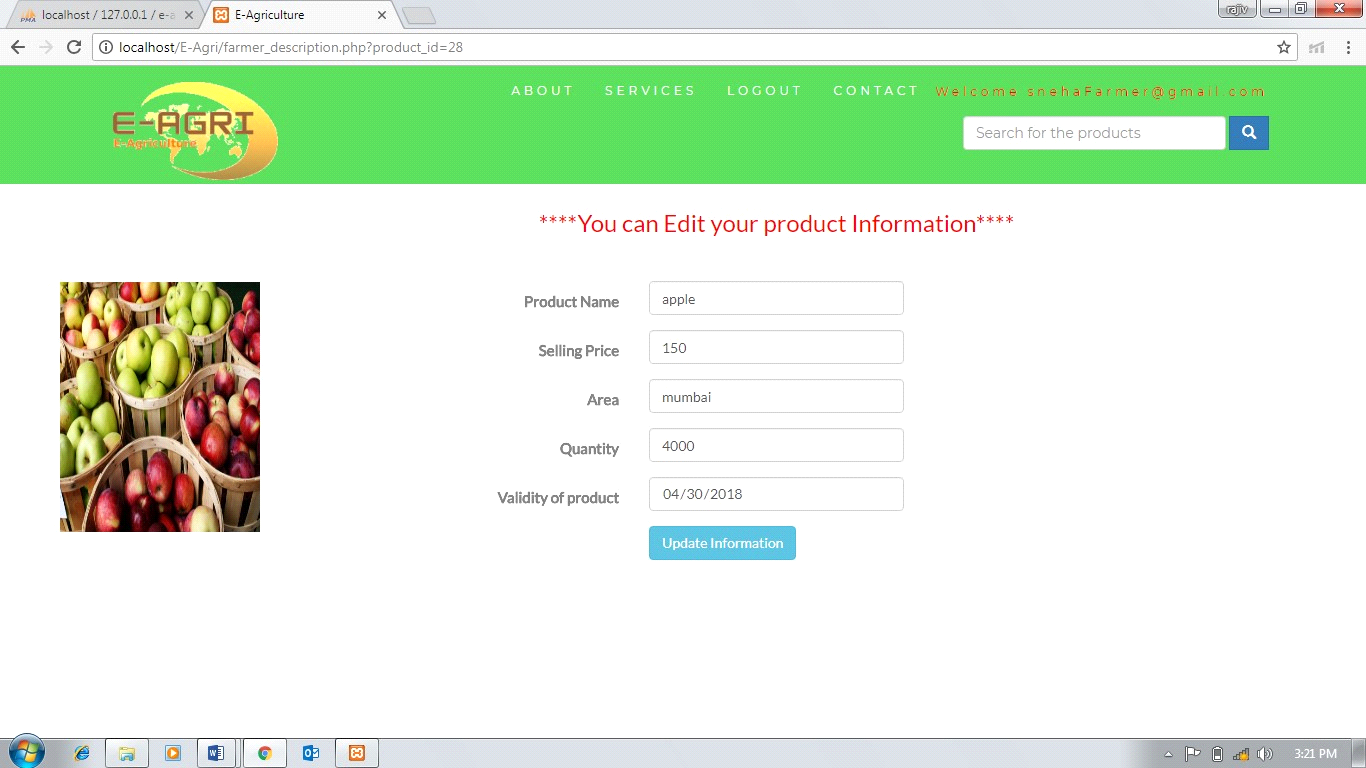
**Payment Page**



**Billing Page or Return Page**

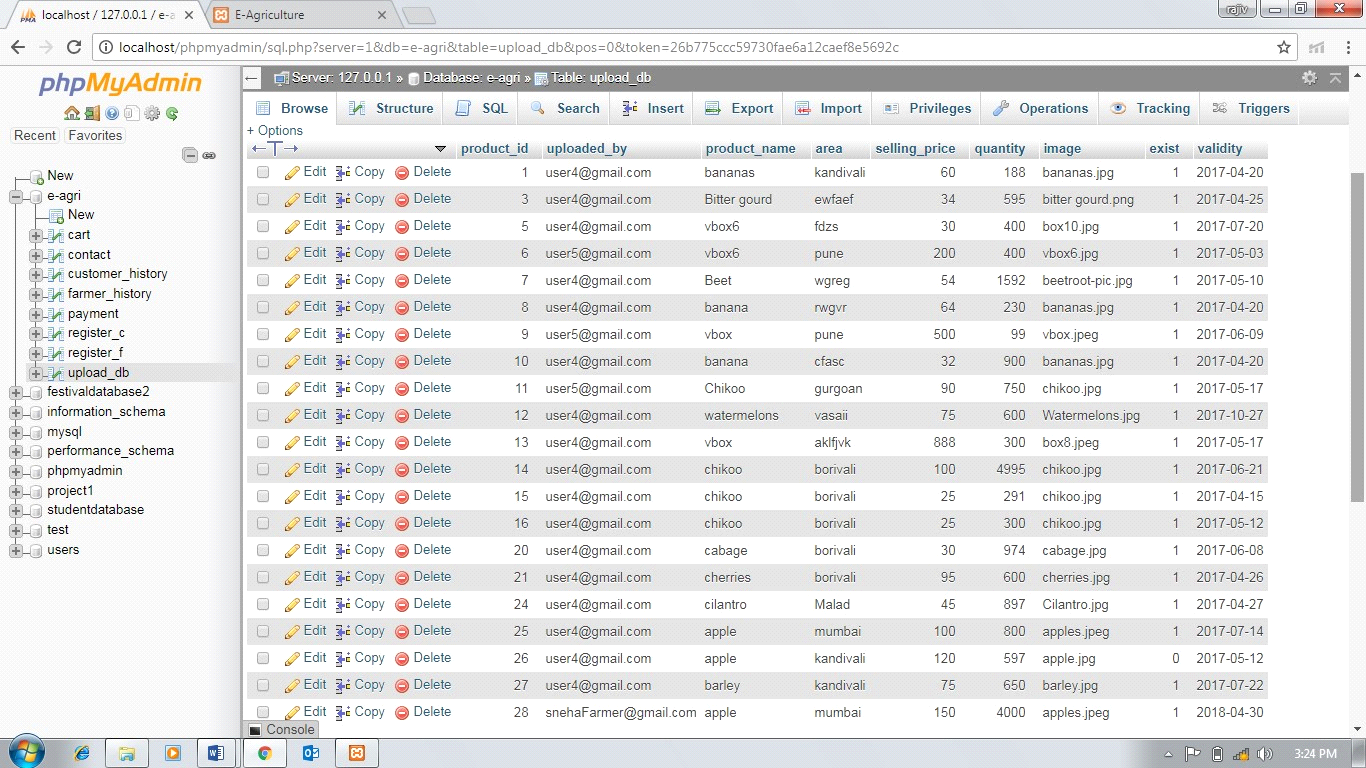


**Update page of product for farmers**

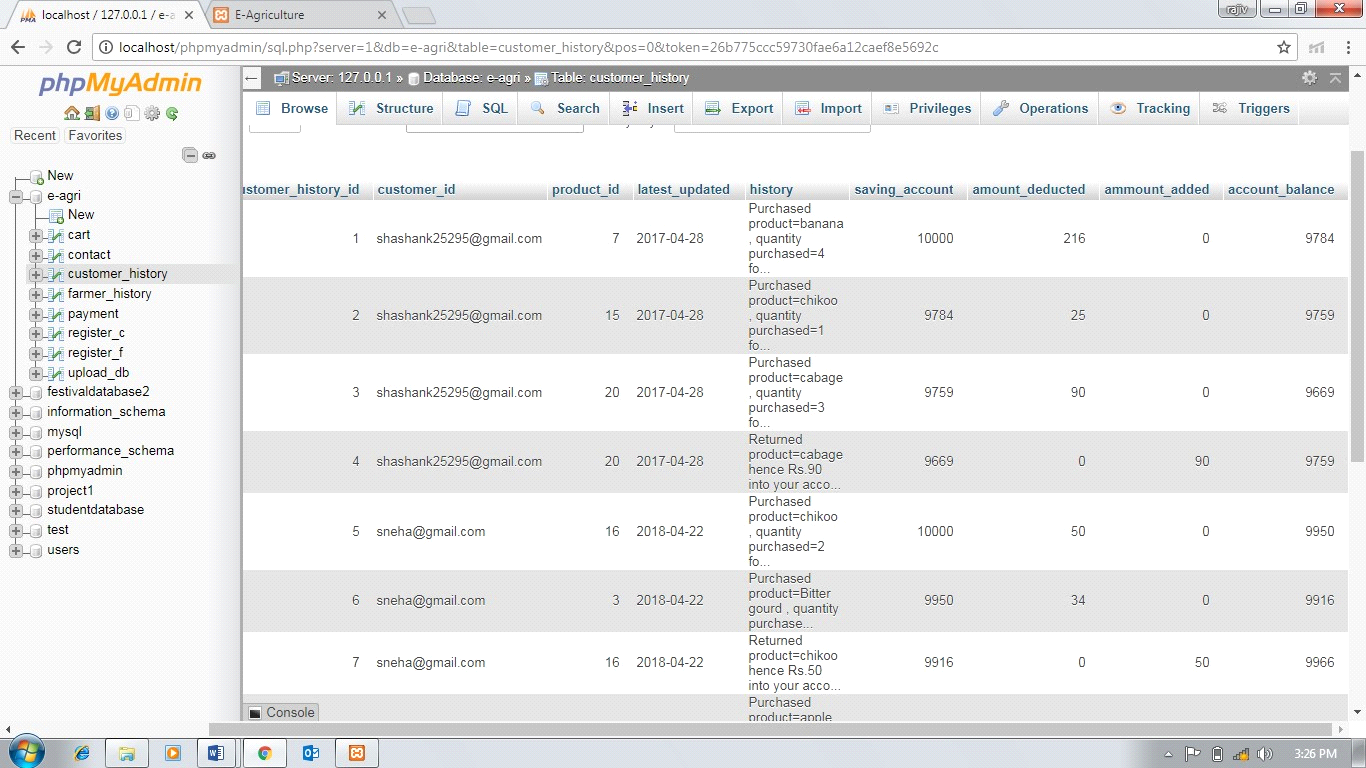


**Screenshot of database tables**

**Product table:**



**Customer History Table:**



**SYSTEM TESTING**

Testing Methods

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

* **White box testing** is when the tester has access to the internal data structures and algorithms including the code that implement these.

\_ Types of white box testing

The following types of white box testing exist:

* API testing (application programming interface) - testing of the application using public and private APIs
* Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
* Fault injection methods - improving the coverage of a test by introducing faults to test code paths
* Mutation testing methods
* Static testing - White box testing includes all static testing

* **Black box testing** treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

**CONCLUSION**

This project is designed to meet the requirement of E- Agriculture system.

It has been developed in BOOTSTRAP,PHP and ORACLE DATABASE

MANAGEMENT SYSTEM keeping in mind the specification of the system

For designing the system we have used simple data flow diagrams.

|  |
| --- |
| . |