INTERNSHIP PROJECT REPORT

(Project Term January-May, 2019)

BioKart

Submitted by:

Vaishali (151400)
Sagar Gaur (151402)
Nikesh Chauhan (151414)
Aastha Gupta (151434)
Abir Soni (151451)
Ekesh Garg (151453)
Dikshit Sharma (151458)

UNDER THE GUIDANCE OF

Mr. Tony Shaji Thomas

System Engineer - Education, Training & Assessment, ETA,

Infosys Limited, Mysore

CERTIFICATE

I hereby declare that the work presented in this report entitled "**BioKart**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of **Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat** is an authentic record of our own work carried out over a period from 5th April 2019 to 17th May 2019 under the supervision of **Mr. Tony Shaji Thomas, Member - Education, Training & Assessment, Infosys Limited, Mysore**.
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Vaishali (151400)

Sagar Gaur (151402)

Nikesh Chauhan (151414)

Aastha Gupta (151434)

Abir Soni (151451)

Ekesh Garg (151453)

Dikshit Sharma (151458)

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Mr. Tony Shaji Thomas

(Member - Education, Training & Assessment, Infosys Limited)

Date: 17/05/2019

ACKNOWLEDGEMENT

We would like to express our gratitude to our project mentor **Mr. Tony Shaji Thomas who** guided us throughout the project **"BioKart".** He helped us in the thorough understanding of project, implementing the functionalities and making our project more responsive and efficient. We would also like to extend our heartiest thanks to **Mr. Arpan Nema**, Member ETA Mysore for his support and guidance throughout the project.

Secondly we would also like to thank our friends and all our mentors who helped us in finalizing the project with in the given time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S.No.	Abbreviation	Definition
1	MVC	Model View Controller
2	DB	Database
3	UI	User Interface
4	ER	Entity Relationship
5	SDLC	Software Development Life Cycle
6	IIS	Internet Information Services
7	DFD	Data Flow Diagram

GRAPHS

S.No.	TITLE
1	Burn Down Chart -Sprint I
2	Burn Down Chart -Sprint II

ABSTRACT

The project aims at developing an E-Commerce website for a firm named **Biokart** that bridges the gap between farmers and customers while acting as an intermediate facility.

The technology that we have chosen to complete the project goals is DOT NET ASP using MVC Architecture for frontend and MSSQL aided by LINQ for database connections and communications. LINQ or Language Integrated Query is used extensively for several CRUD operations like fetching records, creating new auctions, adding bid for auction etc.

The website enables four roles of users – Admin, Employee, Customer and Farmer. While an employee has most of the management and organizational accesses like that of auction, and admin has all these in addition too ability to manage the details of customer. Farmer and customer to get a specially designed dashboard.

CHAPTER 1.

INTRODUCTION

1.1 Introduction

1.1.1 The Model-View-Controller Architecture

The model view Controller architecture gives us an entire responsive application to implement our project. The main features of MVC consist of Model, View and Controller. The model folder is the main data folder that is formed after the process of **Scaffolding** in which we convert the entire tables into an entity class. These gives us the core data for our project to connect to the backend and carry out the functionalities. The Views are used to give the Frontend of the project and the controller are used to manage the functionalities.

Model, View and controller together gives us an interactive application to run a project. The models are used for linking the frontend to backend. The Views are just like a windows to a client to see the frontend of a project. The view may be single razor view or a partial view. The partial view that can be used multiple times.

Every view is linked to a particular controller. The controller inputs the data from the models and represent it to client to the frontend through the views. The users communicate through the controllers. The views created through the controllers calls for a new HTML markup request. The controller also receives the response for all the action that are performed through various functions are created to implement different functionalities.

1.2 Problem Statement

The aim is to create a website a firm **BioKart** that will be used for the following:

- Provide a portal for farmers to advertise their crops/poultry goods.
- Farmers can understand what all products have aspirants but are not yet satisfied by the market.
- Customers can buy fresh organic products in their locality.
- Host auctions for selling of products to facilitate faster selling of goods.
- Fulfil bulk order requests for fruits and vegetables from customers.

1.3 Objectives

Below are the objectives that shall be fulfilled post the implementation of this project:

Farmer Functionalities:

- Should be able to login to their account and update their address details.
- Should be able to specify the type of agricultural/poultry products which are cultivated by them.
- Visibility of the quantity of products available with them as of now should be updated.
- Should be able to see what are the products generally requested by the customers in their locality.

Customers Functionalities:

- Should get a distance sorted suppliers.
- See the details of the farmer supplying the product.
- Place a request on the portal for a particular product.
- Participate in online Auctions hosted by the company.
- Provide a bulk order for a specific date in the future.

Employees Functionalities:

- Manage the login details of farmers and customers.
- Check feedbacks received by the company.
- Analyze the stocks – Regular as well the ones for auction.
- Forward Customer's request for exotic crops to farmers.
- Host online auctions for the farmer's products whenever needed.
- Make sure that the Auction is just and is not facing any technical difficulties.

Admin Functionalities:

- Gets access to all employee functionalities.
- Manage the login details, not only of farmers and customers but also employees.
- Can also enroll new employees to the company's database.

1.4 Methodology

In our project we have made use of MVC architecture that is used for the development of websites. We have used Cshtml, C# for making views and models and controller.

AGILE is a practice of methodology that involve continuous repetition of project maintenance and development as well as improvement throughout the software development lifecycle of the project. Unlike waterfall model in this method, the testing and developing activities are concurrent or it can change as the changes in requirements are seen. Since in many given projects, the requirements are subject to change with the user's demand, or new arrival of module etc.

In this method, the whole cycle of the project is to be divided into different Sprints, where the project can be divided at the end of each sprints, Performance is depicted through the graphs.

1.5 Organization of Report

Chapter 1, highlights and underlines the MVC architecture and framework characteristics. In this chapter we discuss what is MVC and how this architecture can be made use to develop enterprise applications.

Chapter 2 consists of the collaboration and the crux of the studied literature from past few years on the topic.

Chapter 3 covers the evolution of system used and the explanation of the same through diagrams.

The test results, performance analysis are collaborated in **Chapter 4.**

Chapter 5 entails the conclusion of the project hereby proposing some more development in the future scope.

CHAPTER - 2

LITERATURE SURVEY

Audit of writing is a technique to assess and comprehend the looks into with respect to a specific zone. The point in center is to yield a calculative assessment of the theme through different justifiable techniques. The fundamental goal of our study is to locate the important work in regards to MVC design and its different working procedures.

2.1 **"Assessing the Effectiveness of the Model View Controller Architecture for Creating Web Applications", Mike Morrison, Joline Morrison and Nick Heidke.**

The Model View Controller (MVC) design has been broadly acknowledged as a methodology for creating Web-based Enterprise Applications that contains a back-end programming part with a visual yield delivered.

Till now the adequacy of the MVC design for creating different applications has not been tended to. The exploration examined here expects to satisfy this prerequisite by differentiating MVC to numerous other significantly utilized webpage/gathering advancement methods as far as viability, time taken to create and the possibility to improve correspondence among developers and planners by contrasting a MVC-based Web application and a non-MVC Web application and centering the inconveniences and favorable circumstances of each methodology.

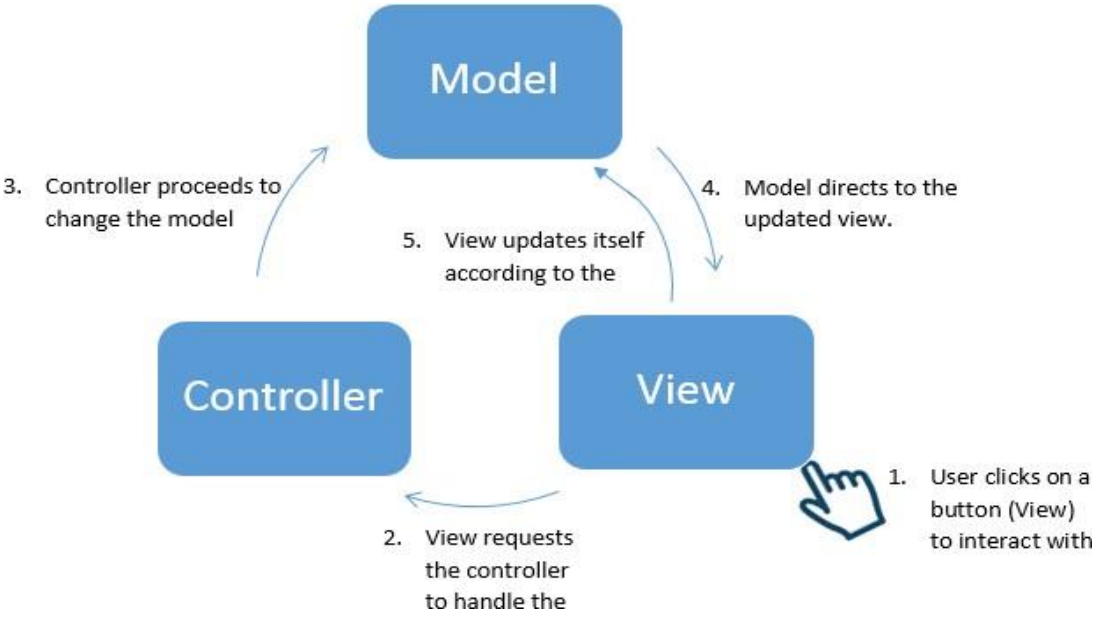


Fig. 1 "MVC Architecture."

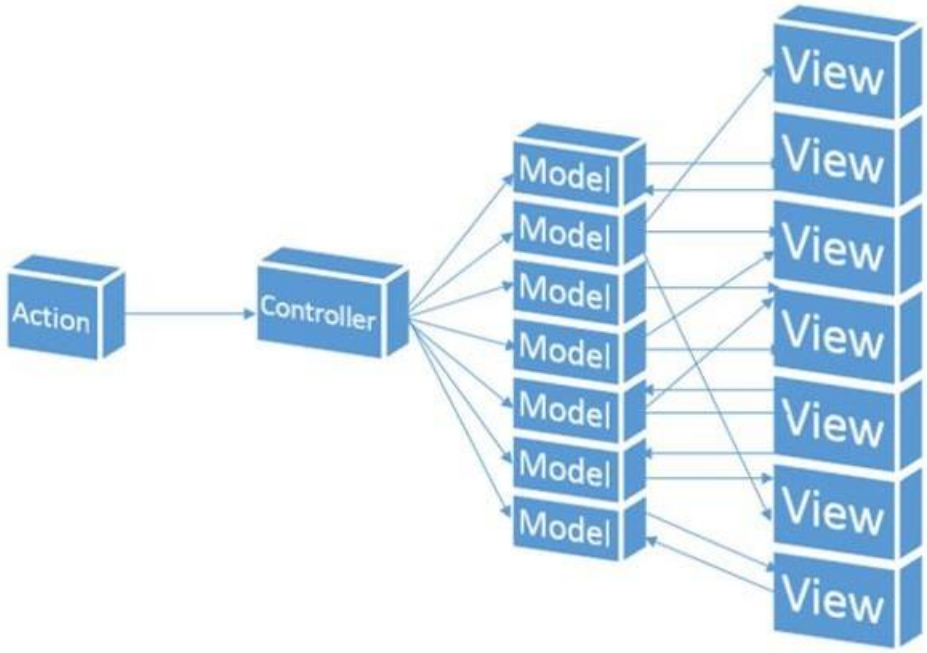


Fig. 2 "MVC Work Flow."

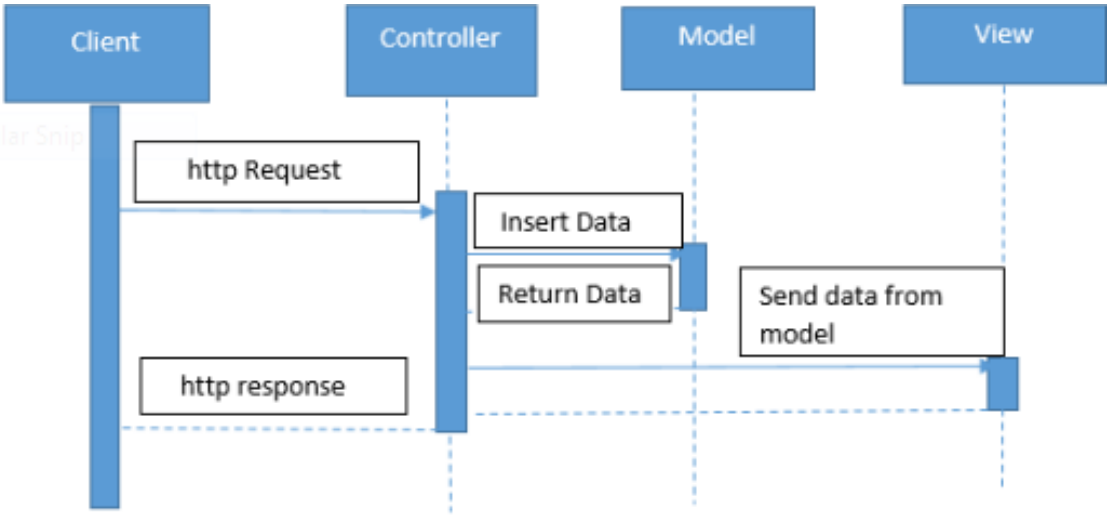


Fig. 3 "MVC Work Flow Activity Diagram."

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Hardware/Software requirement and platform

System Information	Configuration Details
Processor	Intel Pentium 3/5/7
RAM	2GB/4GB/8GB
Operating System	Windows7
HDD	500GB
Speed	1 GHZ and more
Language	C# and CSHTML

Table 1. Hardware & Software Requirements of project.

3.2 Requirement Specifications:

Everyone loves to eat fresh organic fruits and vegetables without any harmful pesticides. People owning green houses and are doing cultivation of organic pure fruits and vegetables in their own fields or backyards finds it difficult to find the right buyers. They are unwilling to go to markets, sit whole day and sell their products directly to customers. Bio Kart is a firm which decides to bridge the gap between organic farmers and organic food aspirants.

The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the portal allow a farmer to keep a record and sell his products and customers to view as well as buy products along with other features such as search.

- **Stock Management: BioKart** allows farmers to update their repository real time. They can add either regular stock or exotic stock exclusively for auctions.
- **Purchase:** Allows customer to buy stock that has been added to his cart or that is available in the farmer's stock. The cart validates the stock real time, allowing customer to purchase only what's available.
- **Auction:** Enables an employee to organize an auction, setting a base bid price and the end date for the auction. Once the end date is exceeded the auction is officially closed and inaccessible while the employee can still access it under "Pass Auctions".
- **Request:** The website gives the flexibility to the customer to request for any product that he wishes to buy but is not currently available in the farmer's stock. The request is accessed by the farmer and if is confirmed the farmer changes the status to committed else the request can be deleted and the concerned customer is notified
- **Feedback Management:** In another attempt to understand the needs and demands of customers, the website enables users to submit feedbacks that can be analyzed by both the admin as well as all the employees and the necessary changes can be made keeping the request, queries and feedback of the customers in mind.

3.2.1 Functional Requirements

(TABLE 2)

1	As a **farmer**, I should be able to login and update my personal details
2	As a **farmer**, I should be able to specify the type of crops cultivated
3	As a **farmer**, I should be able to update stock available and view customer requests
4	As a **customer,** I should be able to view distance organized list of products as per category, along with details of the farmers
5	As a **customer**, I should be able to place requests for a particular product
6	As a **customer**, I should be able to participate in auctions hosted by the company
7	As a **customer**, I should be able to schedule bulk orders
8	As a **customer**, I should be able to filter the products I need
9	As an **Employee**, I should be able to manage the login details of farmers and customers
10	As an **Employee**, I should be able to send notifications to farmers about the general requirements
11	As an **Employee**, I should be able to find the farmers who can take bulk orders
12	As an **Employee**, I should be able to host online auctions when needed
13	As an **Employee**, I should be able to send notification mails to customers regarding auctions

14	As a **customer**, I should be able to register as new user
15	As a **customer**, I should be able to login and update my personal details
16	As a **customer**, I should be able to see a page where all the scheduled auctions are listed
17	As a **customer**, I should be able to send feedback
18	As a **farmer**, I should be able to register as a new user
19	As a **farmer**, I should be able to update the quantity of product available
20	As a **farmer**, I should be able to communicate with the customers and negotiate deals
21	As a **farmer**, I should be able to access the essentials from home page
22	As a **customer**, I should be able to access the essentials from home page
23	As a **farmer**, I should be add the products I want to sell
24	As a **farmer**, I should be able to delete the product
25	As a **farmer**, I should be able to logout and login another profile
26	As a **customer**, I should be able to logout and login another profile
27	As an **Employee**, I should be able to access the database that is used in other tasks
28	As an **Employee**, I should be able to access the essentials from home page
29	As a **farmer**, I should be able to logout and login another profile and get push notifications

3.2.2 Non-Functional Requirements

Attributes of any system or application like its security, performance, reliability, usability are called its nonfunctional requirement. These requirements effect the overall fitness of the application. These nonfunctional requirements are also known as the quality attributes of a system. It shows how the application is supposed to work.

3.2.3 Entity Framework Approach

Approaches to work with Entity Frame Work:

- Code First Approach
- Database First Approach
- Model First Approach

Biokart project has been developed using database first approach that involve the following steps:

- Formulating the database script.
- Scaffolding using dB context in Visual Studio.
- Using LINQ Queries to perform CRUD operations.
- Consuming the data access layer in MVC project

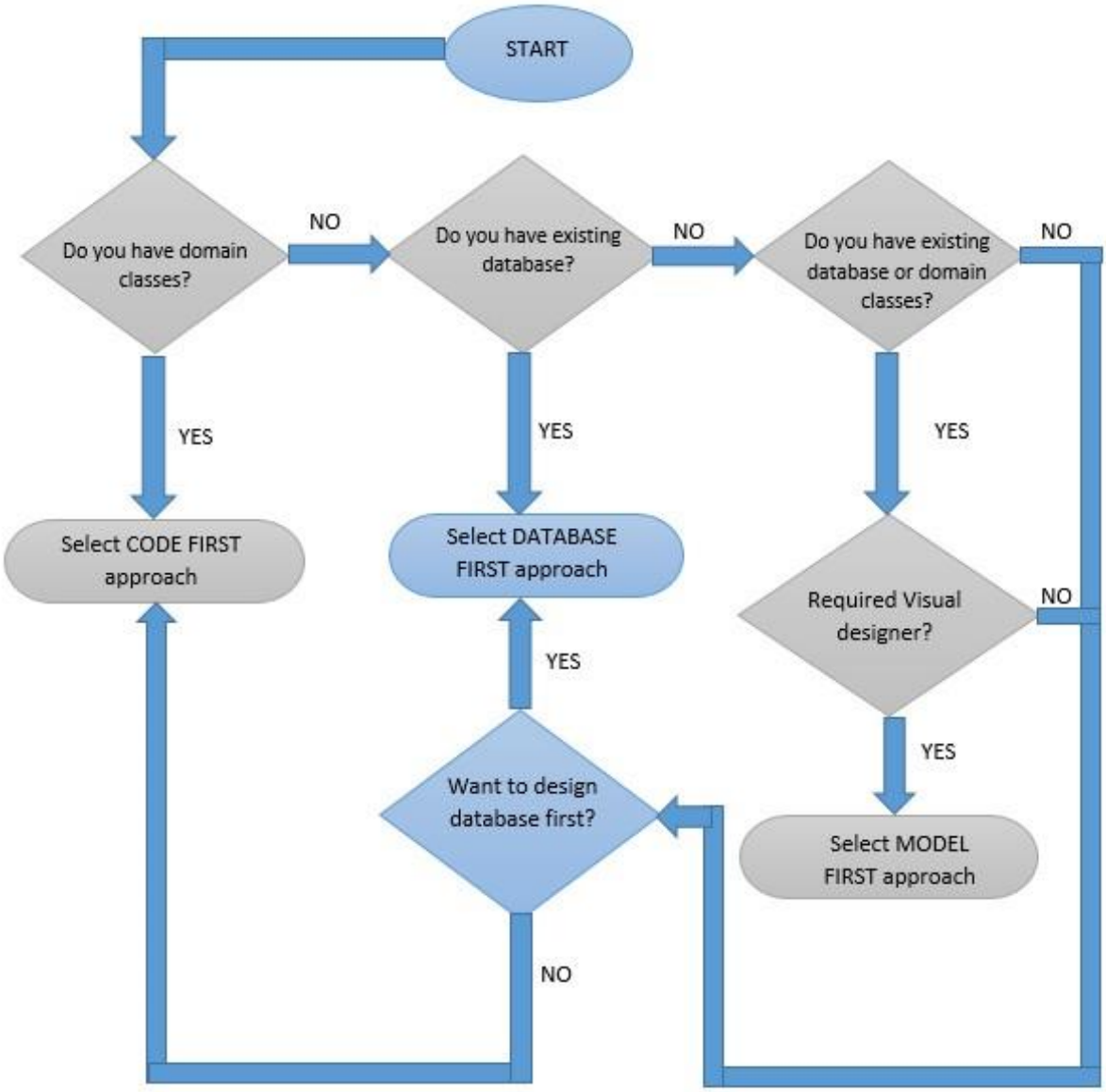


Fig. 4 "Approaches to A Project."

3.3 Data Flow Diagram(DFD):

It is a way in which data flow can be represented. It guides the user about the output and input fields that are used. Different notations are used to display different processes. It shows how information flows from one system to another. To make it more transparent for user, multilevel DFDS are used.

- Flow based methodology.
- Building modules to establish the flow. (In/Out).
- In depth understanding the structure.
- Arrows helps to understand the flow.

3.3.1 Level-0 DataFlowDiagram of The Login Page

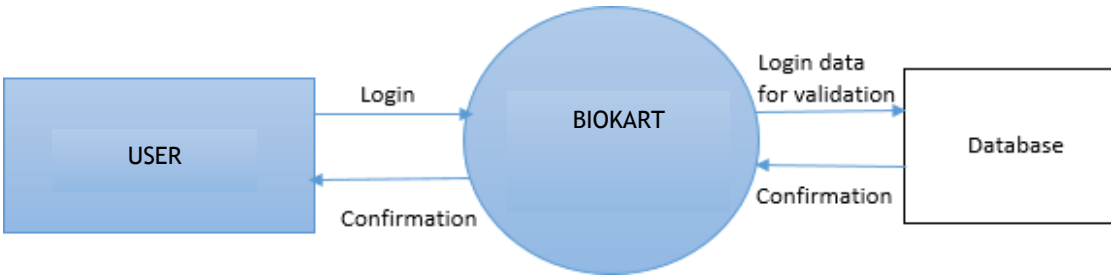


Fig. 5 DataFlowDiagram – Level 0

3.3.2 Level-1 DataFlowDiagram of The Employee Module Wise Flow

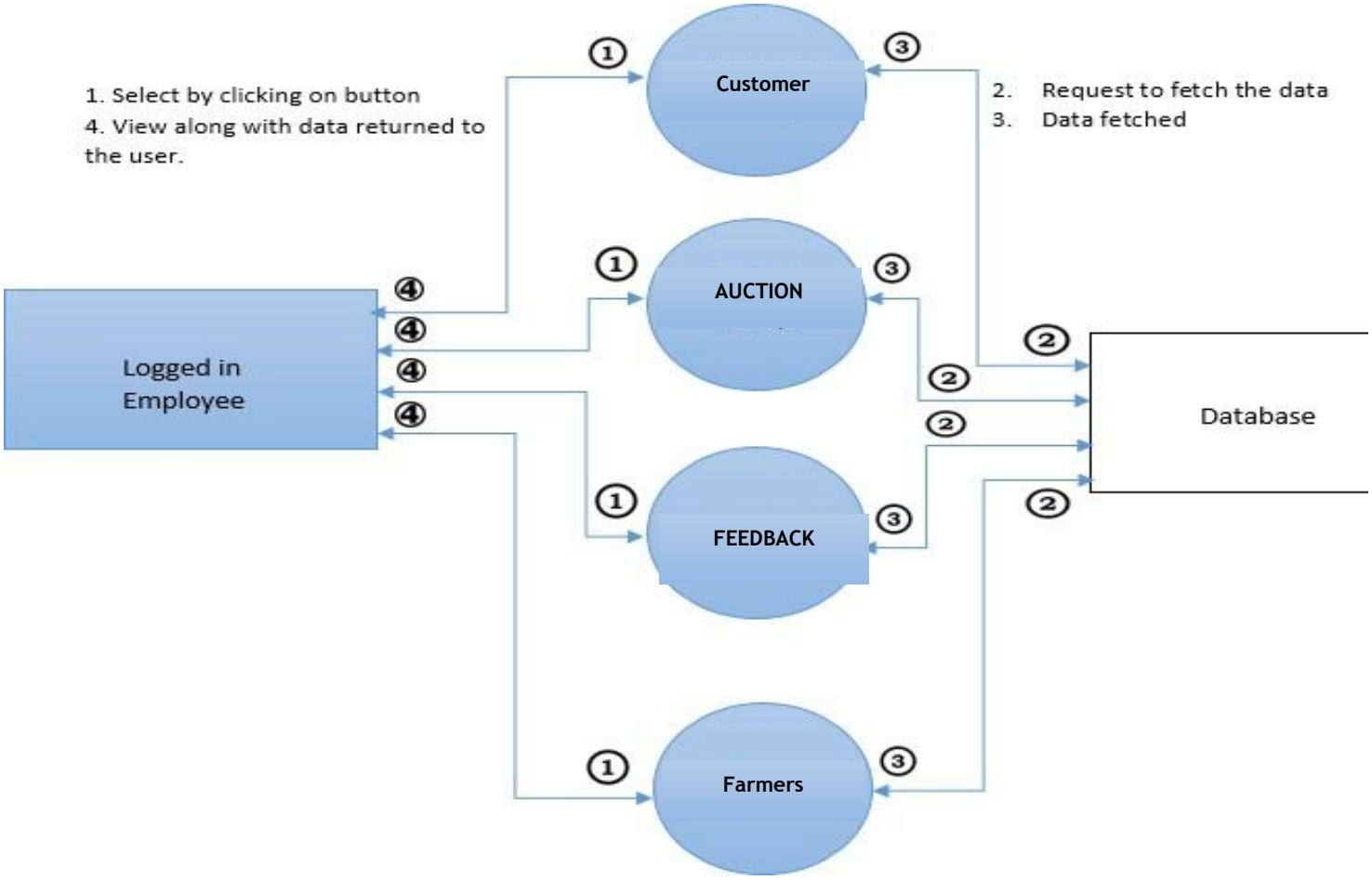


Fig 6. Level-1 DataFlowDiagram; Depicting The Entire Flow for Employee/Admin

3.3.3 Level-1 DataFlowDiagram of The Customer Module Wise Flow

4

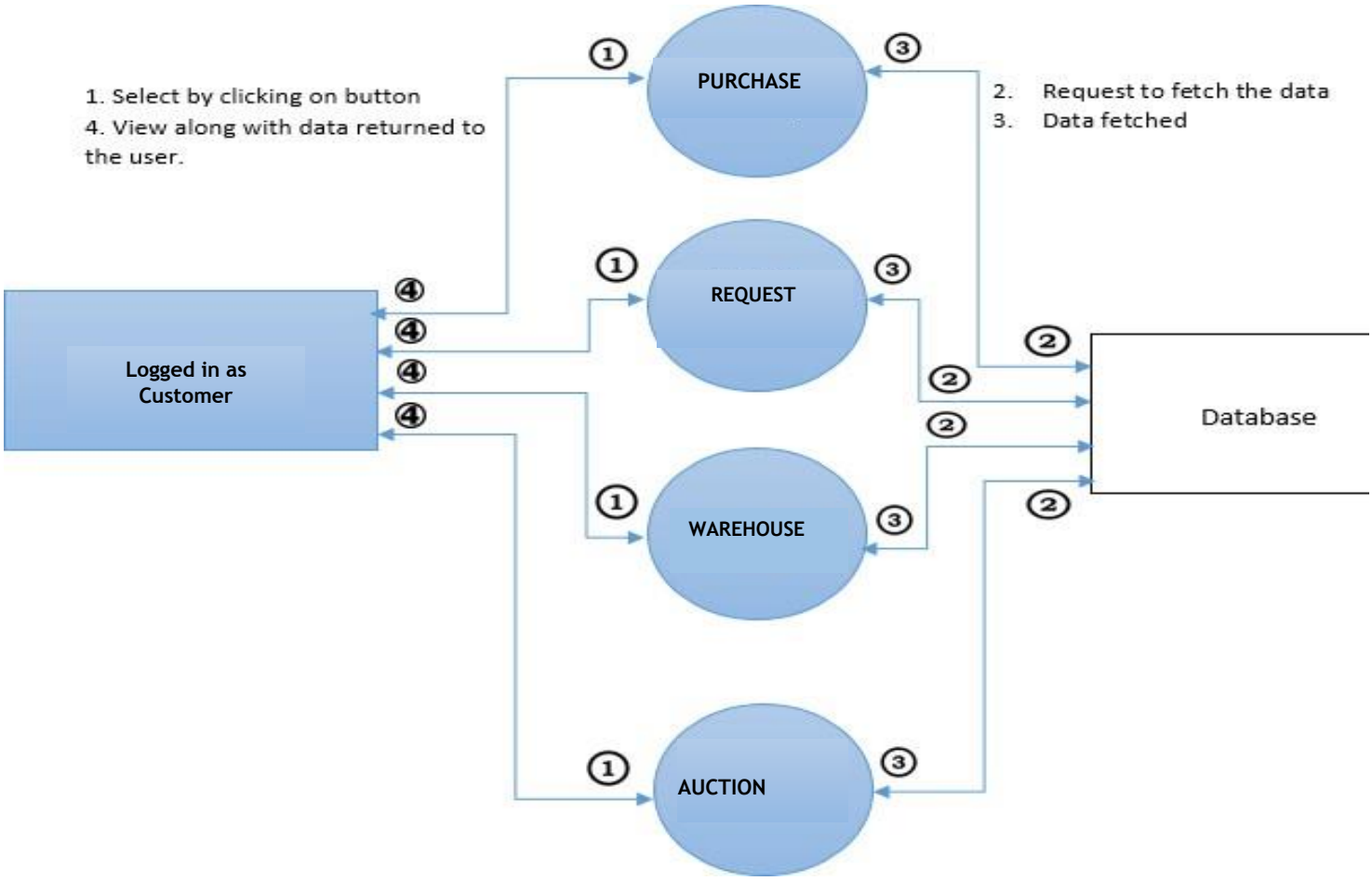


Fig 7. Level-1 DataFlowDiagram; Depicting The Entire Flow for Customer

3.3.4 Level-1 DataFlowDiagram of The Farmer Module Flow

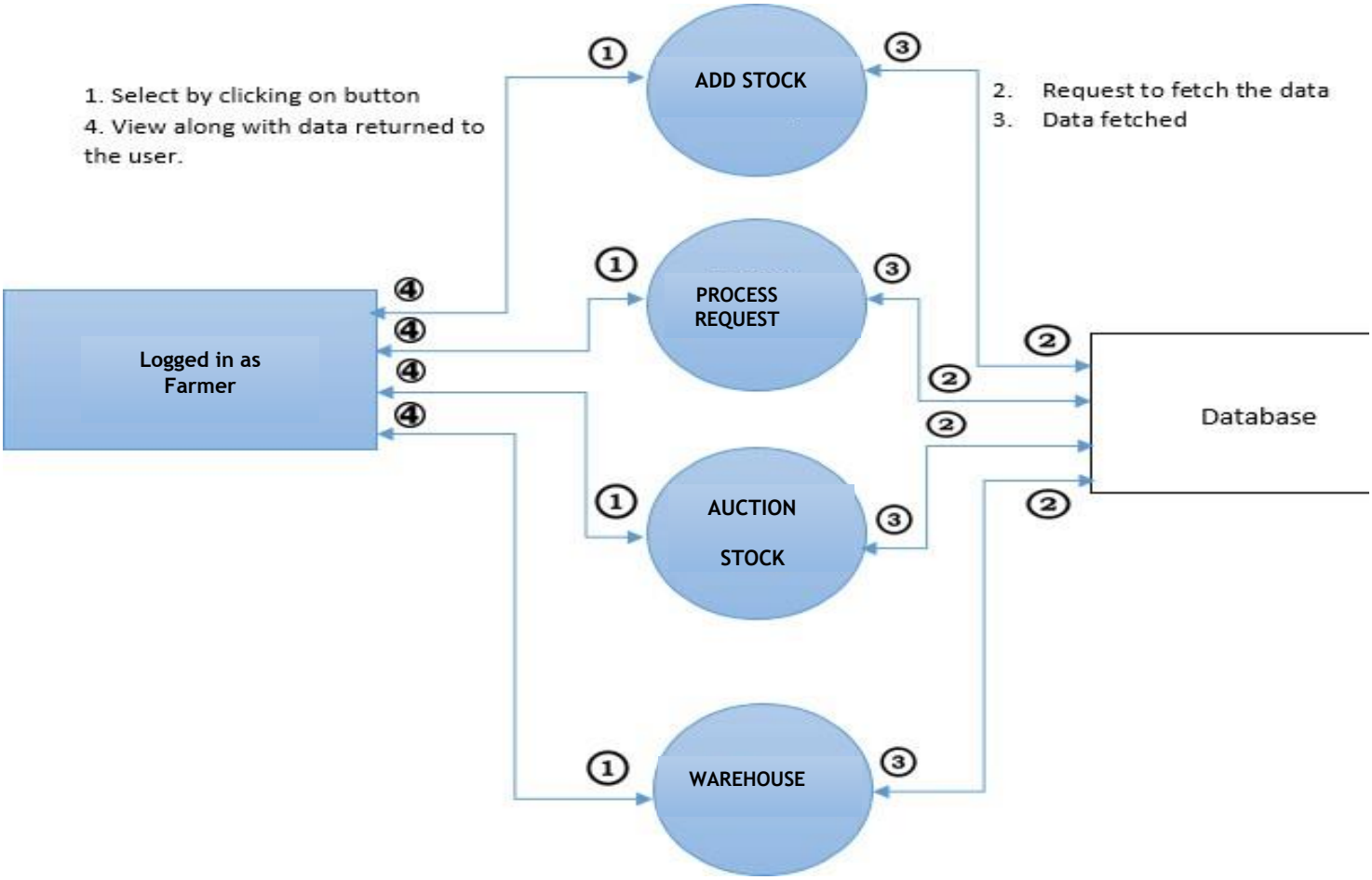


Fig 8. Level-1 DataFlowDiagram; Depicting The Entire Flow for Farmer

3.4 Use Case Diagram

Use case diagrams depicts the relationship between the users and the available functionality. It helps to understand the individual modules functioning as well as the entire system.

Users (actors) of BIOKART include employee/admin, customer, farmer. The use cases have been described below:

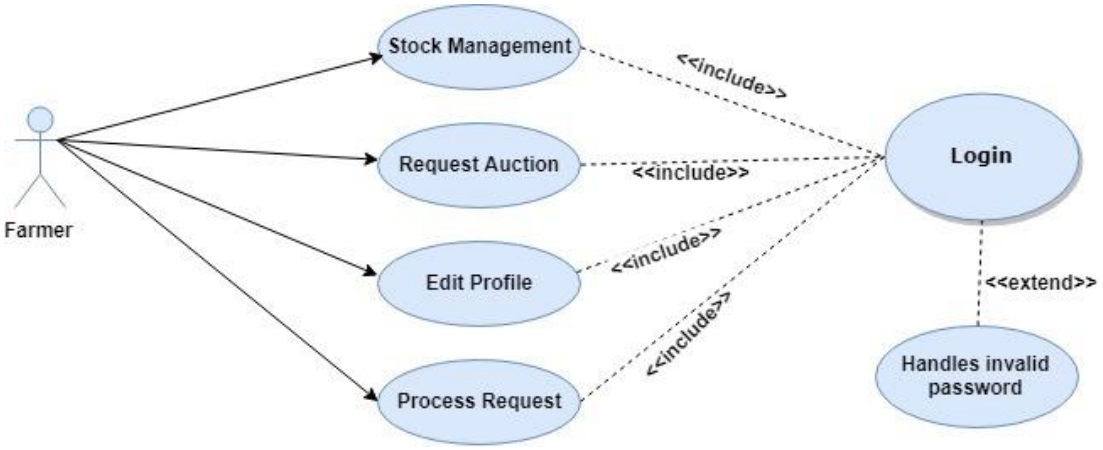


Fig 9. Use Case for Farmer

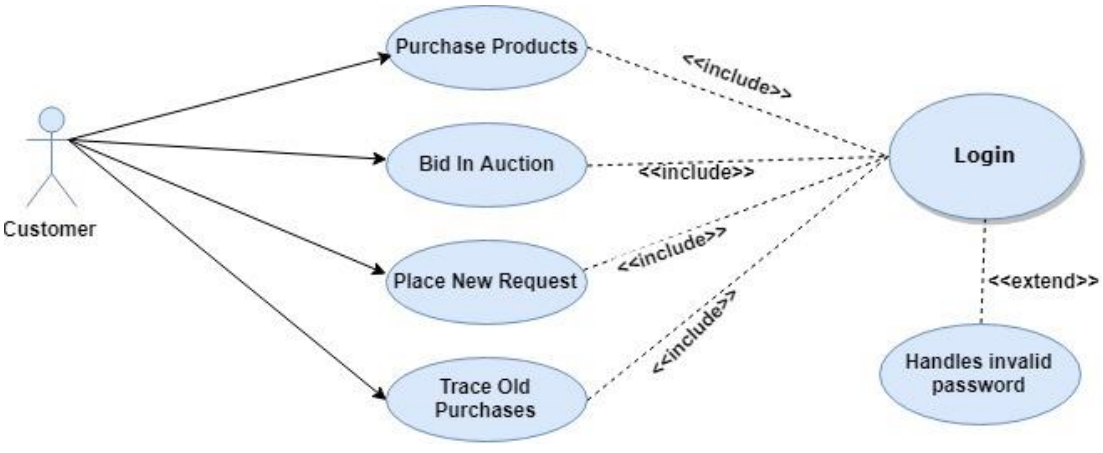


Fig 10. Use Case for Customer

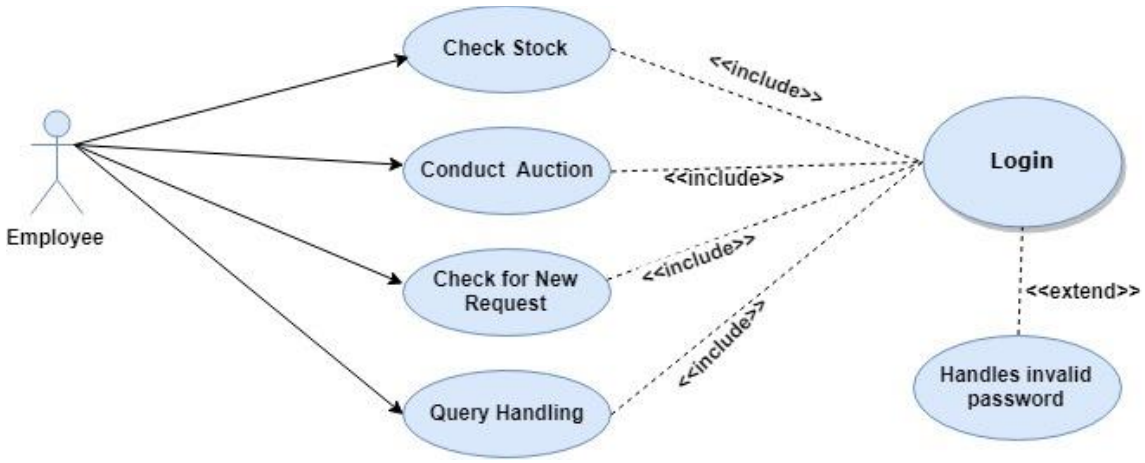


Fig 11. Use Case for Employee

3.5 HIGH LEVEL DESIGN ARCHITECTURE

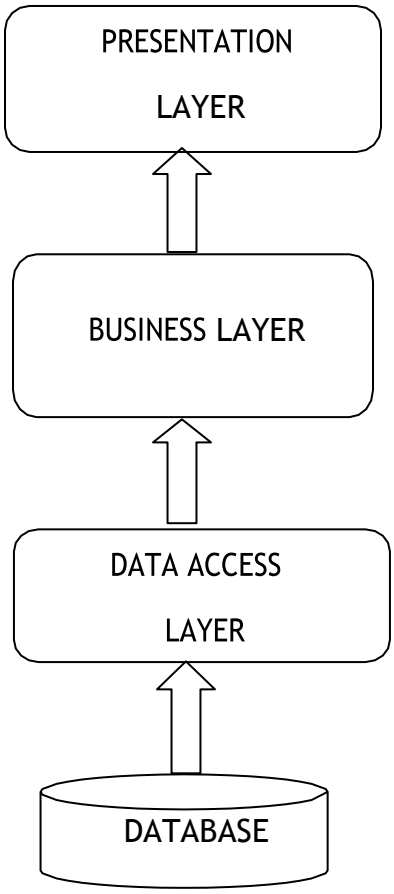


Fig 12. "Design Architecture"

3.6 DESIGINING OF DATABASE

(TABLE NO. 3)

3.6.1 Roles

Field Name	Field Type	Data Type	Mandatory	Possible Values
RoleID	Char (1)	Alphabetic	Yes	RoleId
RoleName	Varchar(20)	Alphabetic	Yes	RoleName

3.6.2 Users

Field Name	Field Type	Data Type	Mandatory	Possible Values
Name	Varchar(100)	Alphabetic	Yes	Name
UID	Integer Identity(100,1)	Numeric	Yes	User ID
Email ID	Varchar(50)	Alphanumeric	Yes	Email id
User Password	Varchar(15)	Alphanumeric	Yes	Password
Role ID	Char(1)	Alphabetic	Yes	Role id
PAN	Varchar(10)	Alphanumeric	Yes	Pan no.
Phone no	Big Integer	Numeric	Yes	Phone no
Address	Varchar(200)	Alphanumeric	Yes	Address
Security	Char(100)	Alphanumeric	Yes	security
Pin	Varchar(6)	Numeric	Yes	Pin code

3.6.3 CATEGORIES

Field Name	Field Type	Data Type	Mandatory	Possible Values
Category Id	Integer	Numeric	Yes	Category id
Category Name	Varchar(20)	Alphanumeric	Yes	Category Name

3.6.4 FarmerStock

Field Name	Field Type	Data Type	Mandatory	Possible Values
UID	Integer	Numeric	Yes	UID
Quantity	Integer	Alphanumeric	Yes	Quantity
Item	Varchar(50)	Alphanumeric	Yes	Item type
Price Per Unit	Numeric(10,2)	Numeric	Yes	Price per unit

PinCode	Varchar(6)	Numeric	Yes	PinCode
Category Id	Integer	Numeric	Yes	Category id

3.6.5 PURCHASE DETAILS

Field Name	Field Type	Data Type	Mandatory	Possible Values
Buyer	Integer	Numeric	Yes	Buyer
Seller	Integer	Numeric	Yes	Seller
Name	Varchar(15)	Alphabetic	Yes	Name
Purchase Type	Char(1)	Alphabetic	Yes	Purchase type
Purchase Id	Integer	Numeric	Yes	Purchase id
Delivery Date	Date	DD/MM/YYYY	Yes	Delivery date
Ordered Date	Date Time	Timestamp	Yes	Ordered date
Item Name	Varchar(50)	Alphanumeric	Yes	Item Name
Quantity Purchased	Integer	Numeric	Yes	Quantity purchased
Price per unit	Numeric	Numeric	Yes	Price per unit
Total Amount	Numeric	Numeric	Yes	Total Amount

3.6.6 AuctionStock

Field Name	Field Type	Data Type	Mandatory	Possible Values
UID	Integer	Numeric	Yes	UID
Quantity	Integer	Alphanumeric	Yes	Quantity
Item	Varchar(50)	Alphanumeric	Yes	Item type
Price Per Unit	Numeric(10,2)	Numeric	Yes	Price per unit
Category Id	Integer	Numeric	Yes	Category id

3.6.7. CART

Field Name	Field Type	Data Type	Mandatory	Possible Values
Cart Id	Integer	Numeric	Yes	Cart Id
Buyer	Integer	Numeric	Yes	Buyer
Seller	Integer	Numeric	Yes	Seller
Price per unit	Numeric	Numeric	Yes	Price per unit
Item Name	Varchar(50)	Alphanumeric	Yes	Item Name

3.6.8. FEEDBACK

Field Name	Field Type	Data Type	Mandatory	Possible Values
Email Id	Varchar(50)	Alphanumeric	Yes	Email id
Phone no.	Big integer	Numeric	Yes	Phone no
Description	Varchar(200)	Alphanumeric	Yes	description

3.6.9. REQUESTS

Field Name	Field Type	Data Type	Mandatory	Possible Values
CID	Integer	Numeric	Yes	Customer id
Quantity	Integer	Numeric	Yes	Quantity
Item	Varchar(50)	Alphabetic	Yes	Item
RID	Integer	Numeric	Yes	Role id
Forward Status	Char(1)	Alphabetic	Yes	Forward status

3.6.10. ADMIN FORWARDED REQUEST

Field Name	Field Type	Data Type	Mandatory	Possible Values
CID	Integer	Numeric	Yes	Customer id
CustomerRID	Integer	Numeric	Yes	CustomerId
Quantity	Varchar(50)	Alphabetic	Yes	Quantity
Item	Varchar()	Alphabetic	Yes	Item
ARId	Char(1)	Alphabetic	Yes	Admin ForwardedRequestId
FarmerId	Integer	Numeric	Yes	FarmerID
Status	Char(1)	Numeric	Yes	Status

3.6.11. NOTIFICATIONS

Field Name	Field Type	Data Type	Mandatory	Possible Values
UserId	Integer	Numeric	Yes	UserId
Description	Varchar(100)	Numeric	Yes	Description
FarmerId	Integer	Alphabetic	Yes	FarmerId
Nid	Integer	Numeric	Yes	NotificationId
Created	DateTime	DD/MM/YYYY	Yes	Notifications

3.6.12. AUCTION ITEM

Field Name	Field Type	Data Type	Mandatory	Possible Values
SellerId	Integer	Numeric	Yes	Customer id
FinalQuantity	Integer	Numeric	Yes	Quantity
Item	Varchar(50)	Alphabetic	Yes	Item
BasePrice	Integer	Numeric	Yes	Role id
StartDate	Date	DD/MM/YYYY	Yes	Forward status
EndDate	Date	DD/MM/YYYY	Yes	EndDate
AuctionId	Integer	Integer	Yes	AuctionId
EmpId	Integer	Integer	Yes	EmpID
AuctionStatus	Varchar(50)	Varchar	Yes	AuctionStatus

3.6.13 AUCTIONBID

Field Name	Field Type	Data Type	Mandatory	Possible Values
AuctionId	Integer	Numeric	Yes	AuctionId
BidderID	Integer	Numeric	Yes	BidderID
BID	Integer	Numeric	Yes	BidId
BidDate	Date	DD/MM/YYYY	Yes	Bidding Date
Bid Amount	Numeric	Numeric	Yes	BidAmount

3.6.14 PAST AUCTION RESULT

Field Name	Field Type	Data Type	Mandatory	Possible Values
AuctionId	Integer	Numeric	Yes	AuctionId
WinnerId	Integer	Numeric	Yes	WinnerId
FarmerId	Integer	Numeric	Yes	FarmerId
BidAmount	Numeric	Numeric	Yes	Bidding Amount
EndDate	Date	DD/MM/YYYY	Yes	EndDate
AuctionResult	Integer	Numeric	Yes	AuctionResult

CHAPTER 4

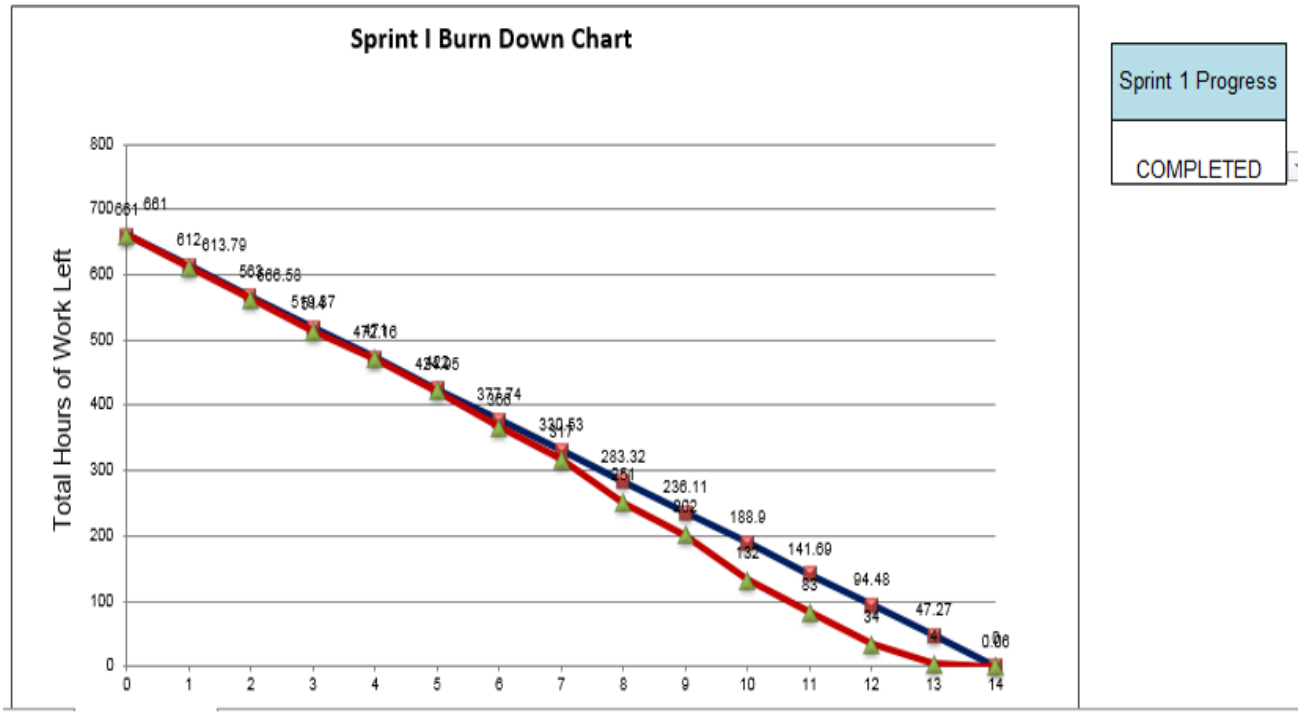
ANALYSIS OF PERFORMANCE

4.1Agile Methodology

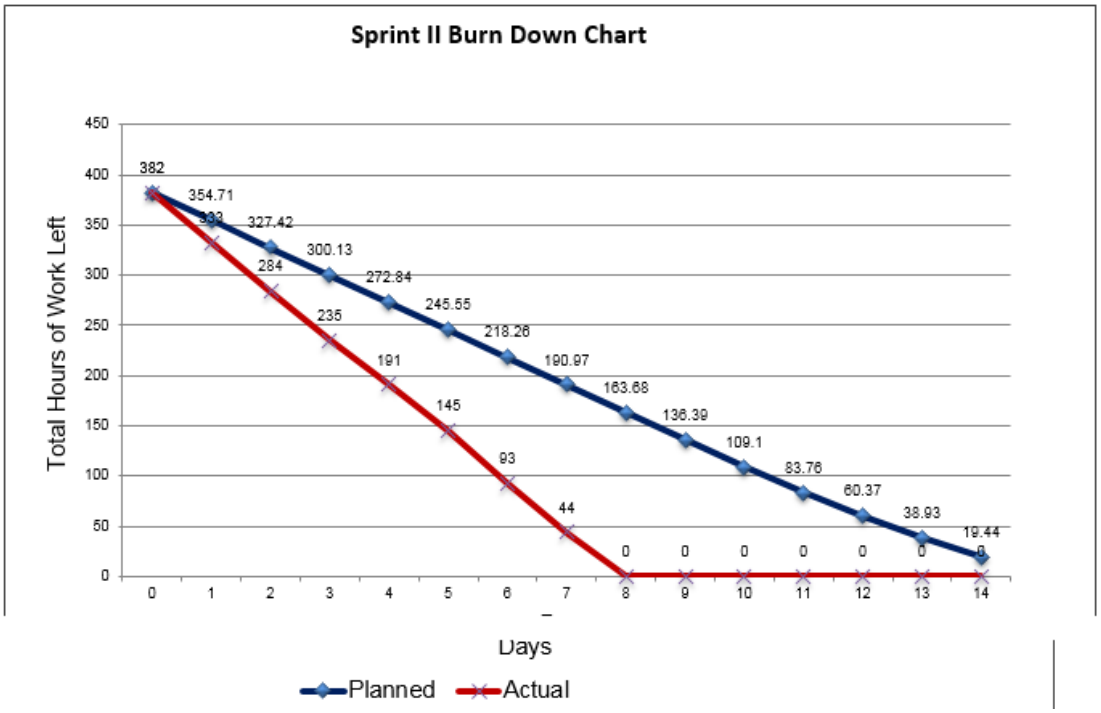
AGILE methodology holds a great profit in application over other modes like waterfall. The reason is ability of AGILE model to allow changes, even when the basic project is ready. It allows rigorous improvement to both the UI and as well as back ends, since it uses the Sprint flow. This means that rather than formulating the project plan in one go, multiple development cycles are designed to helps incorporate the last, minutes changes.

Fig 13. Software Development Cycle

The BIOKART project too has been formulated on this model. While Sprint 0 gave the project the basis planning and database design, the sprint I and II involved development of purchase and option features along with the basics requirements.



Graph1. Sprint I "Burn Down Chart"



Graph2. Sprint II "Burn Down Chart"

(Completed in 8 Days)

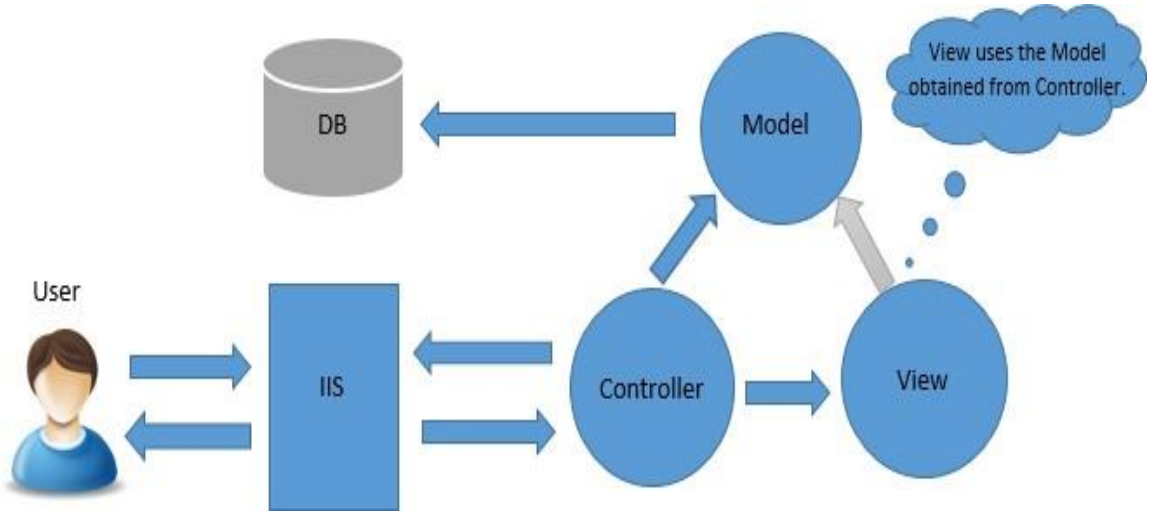


Fig 14. Software Development Cycle

4.2 Test Plan

Test plan is a document or report that gives the description of the strategy of the plan, the objectives of testing, software and hardware required for testing that includes all the resources that are needed for testing. It helps in testing or verifying the ability of the application that you have created. It works as a book of rules and guides us throughout.

4.2.1 Functionality Testing Of Login

Sr.No.	Case	Expected Fields	Actual	Improved
1.	Farmer User Registration	Email ID, Phone, password, PinCode and other details to be validated	Validation not implemented	Validation added
2	Customer User Registration	Email ID, Phone, password, PinCode and other details to be validated	Validation not implemented	Validation added
3	Employee User Registration	Email ID, Phone, password, PinCode and other details to be validated	Validation not implemented	Validation added
4	Admin employee View	Should not be available to the employee.	Employee dashboard included emp view.	Admin home separated from employee home

5	User logged in and clicks on signup	Redirects to his dashboard	Login in Page	“Already logged in?” If yes then dashboard
6	User logged in and clicks on signup	Back to login	Sign Up page	“Already logged in?” If yes then dashboard
7	User Signed in And uses a different role URL/ link	Redirects to his dashboard	Gets unauthorized access	“Already logged in?” If yes then same page

Table 4. Hardware and Software Requirements.

CHAPTER 5

CONCLUSION

After completion of this project we have concluded that this website works as per the need and requirement of the client and is user friendly. All the bugs and errors are thoroughly debugged and thoroughly tested properly. All utility services are successfully integrated at one place in the application. This deploys the kind of application that user want in present time.

5.1 FUTURE SCOPE

Like Every Other Project this project also has a huge scope for improvement but accomplishing those would have been difficult in the given stipulated time. Thus in future the application can be further improved by inculcating some of the features like:

1. Payment portal Integration
2. API Service Layer
3. Automated Auction
4. Image View for the Product
5. Google API for efficient delivery information

DEMONSTRATION

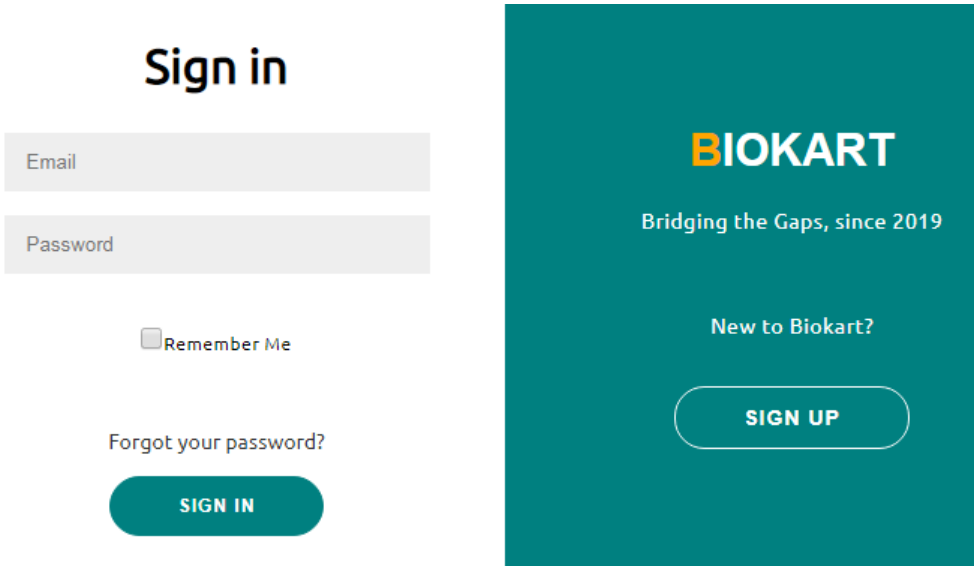


Fig 15. Login Page

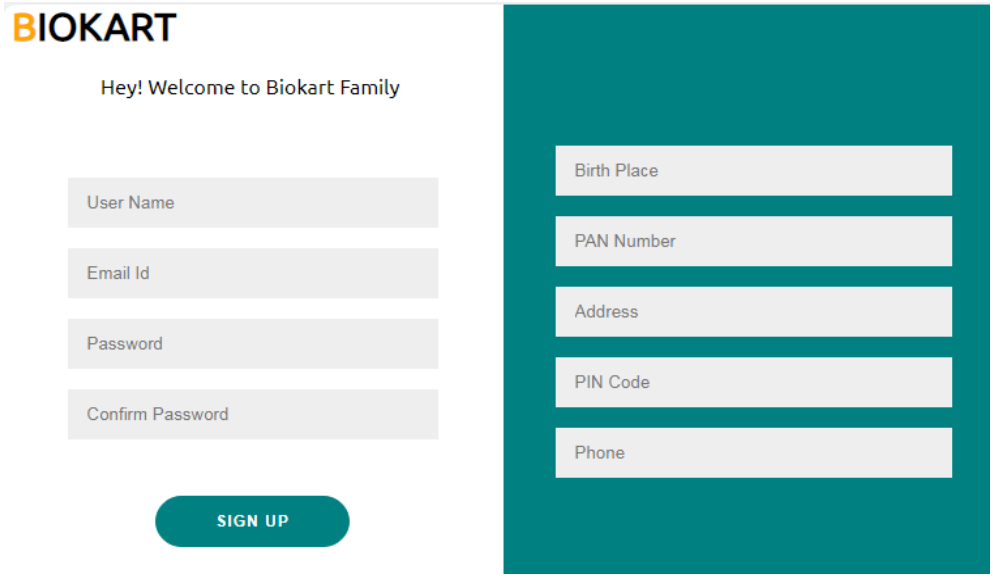


Fig 16. Sign in Page for new user

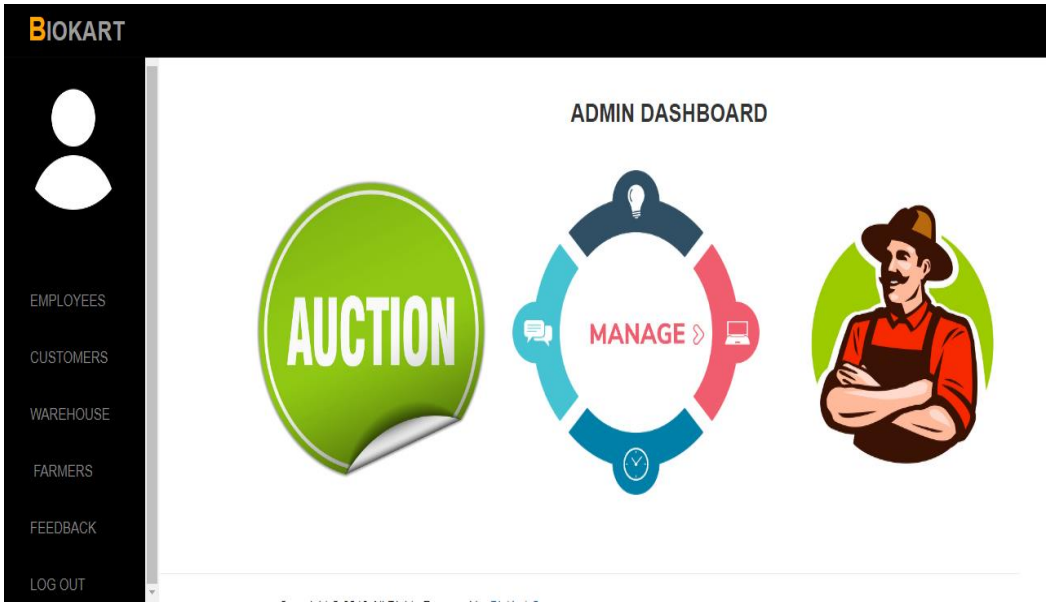


Fig 17. Admin Dashboard

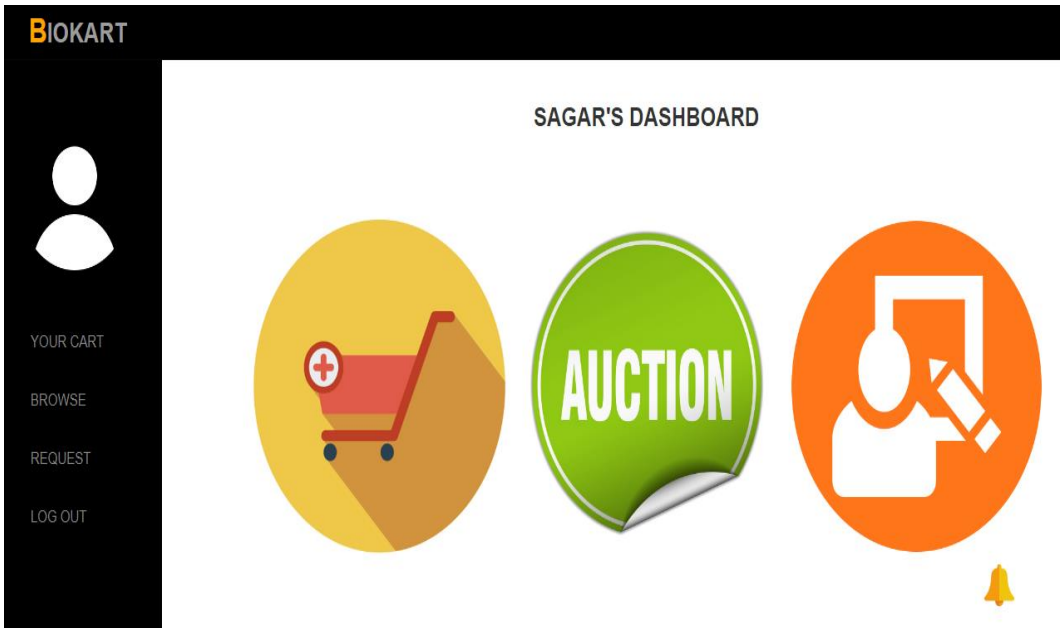


Fig 18. Customer Dashboard

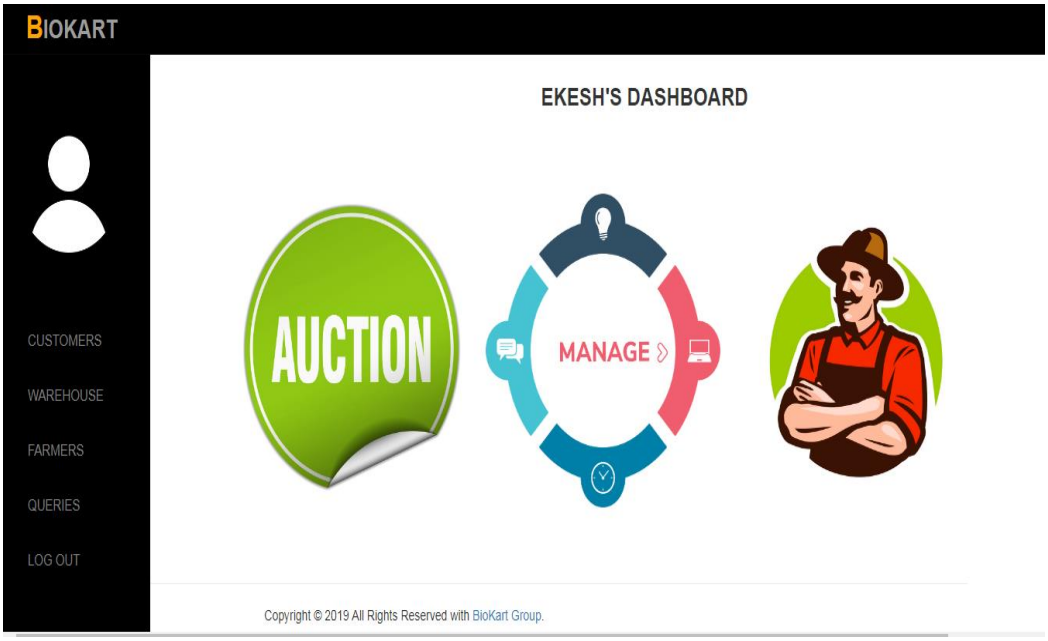


Fig 19. Employee Dashboard

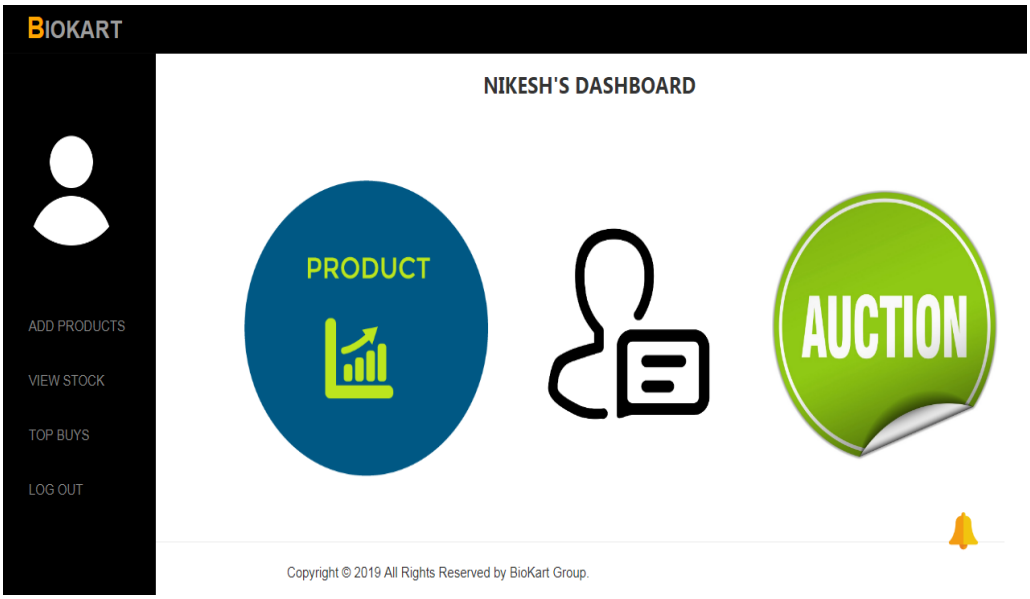


Fig 20. Farmer Dashboard

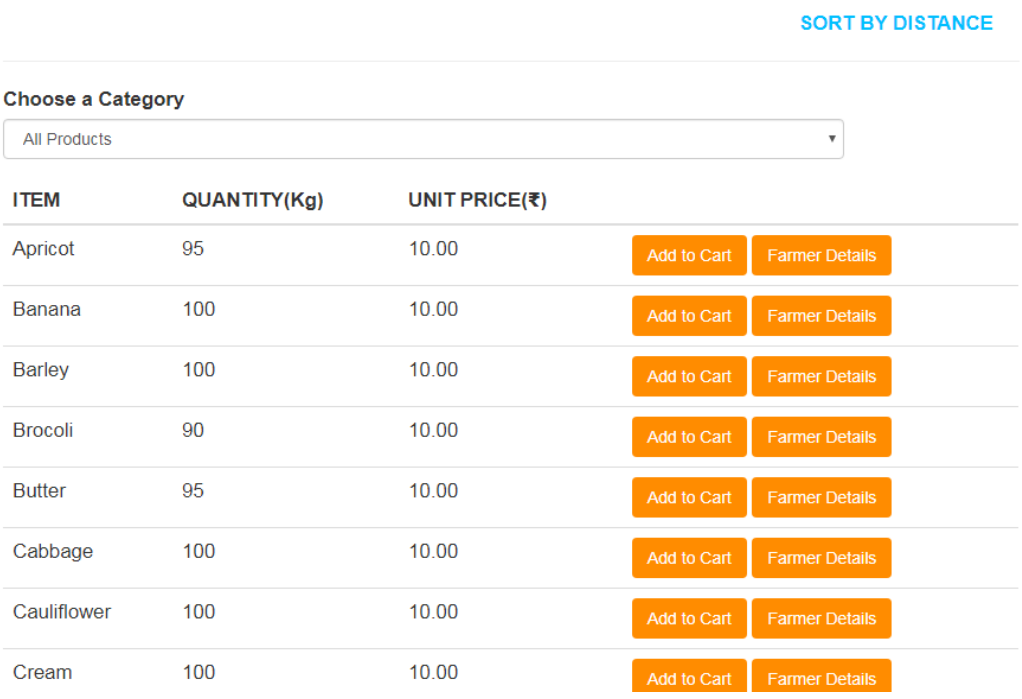


Fig 21. Warehouse - Biokart

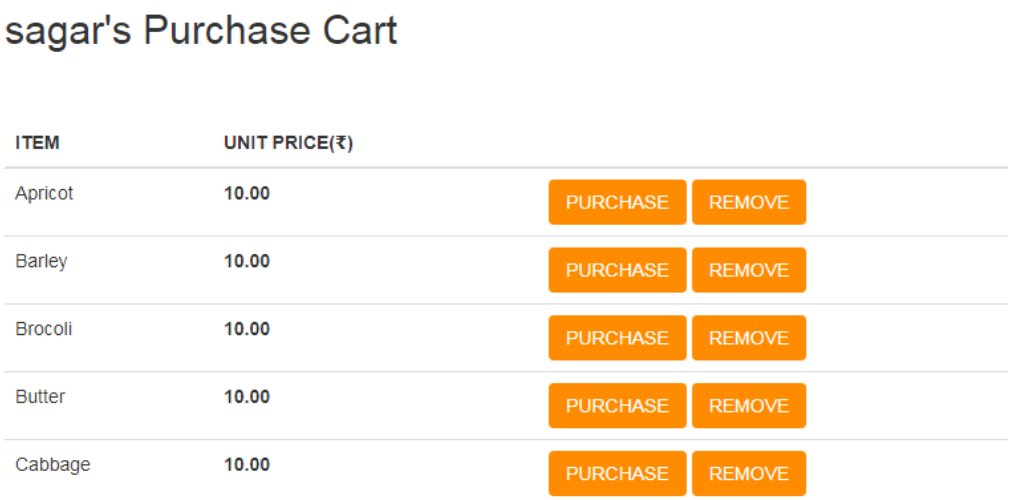


Fig 22. Customer Purchase Cart

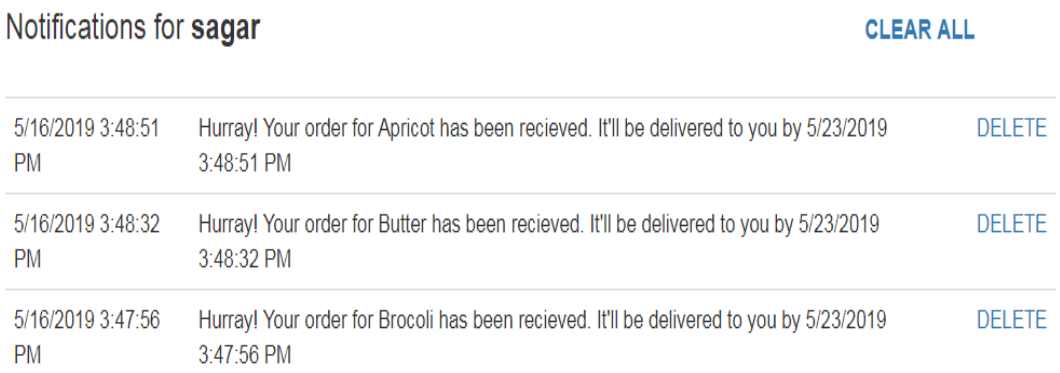


Fig 23. Customer notified when item is purchased

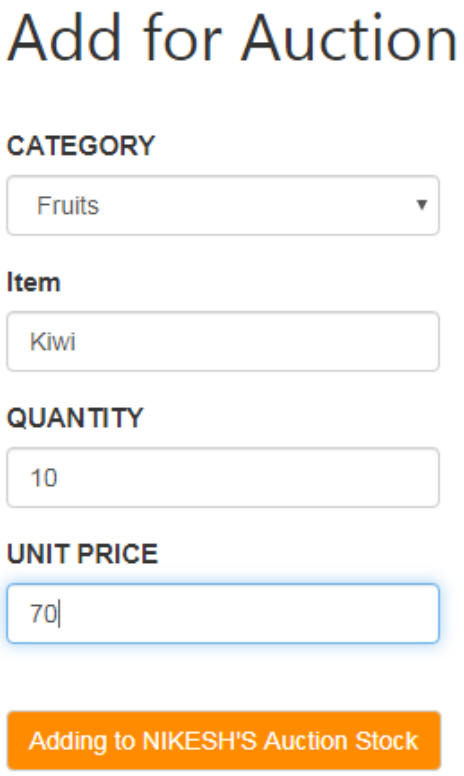


Fig 24. Farmer can add item for Auction

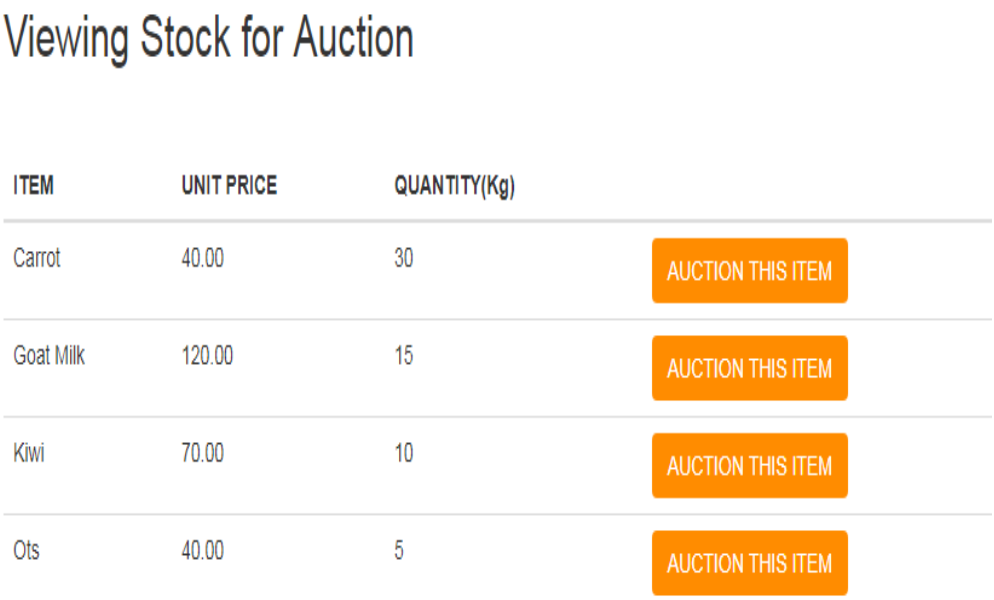


Fig 25. Employee forward the request to all the customer

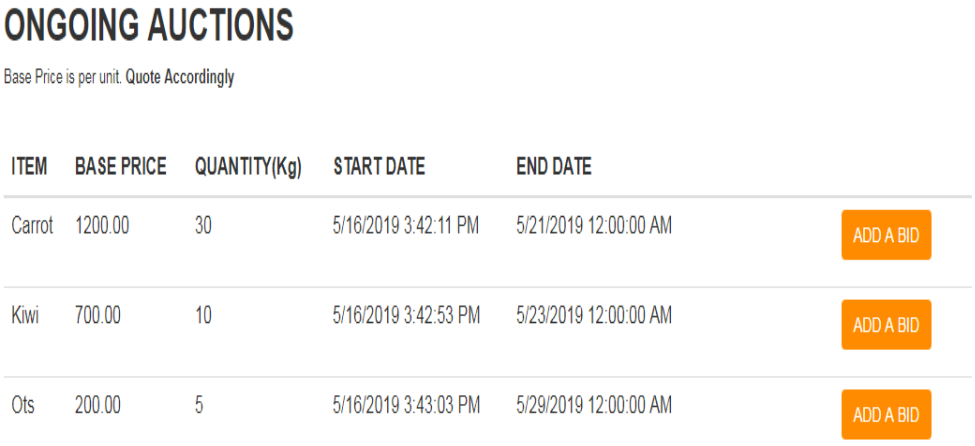


Fig 26. Customer can add bid for item

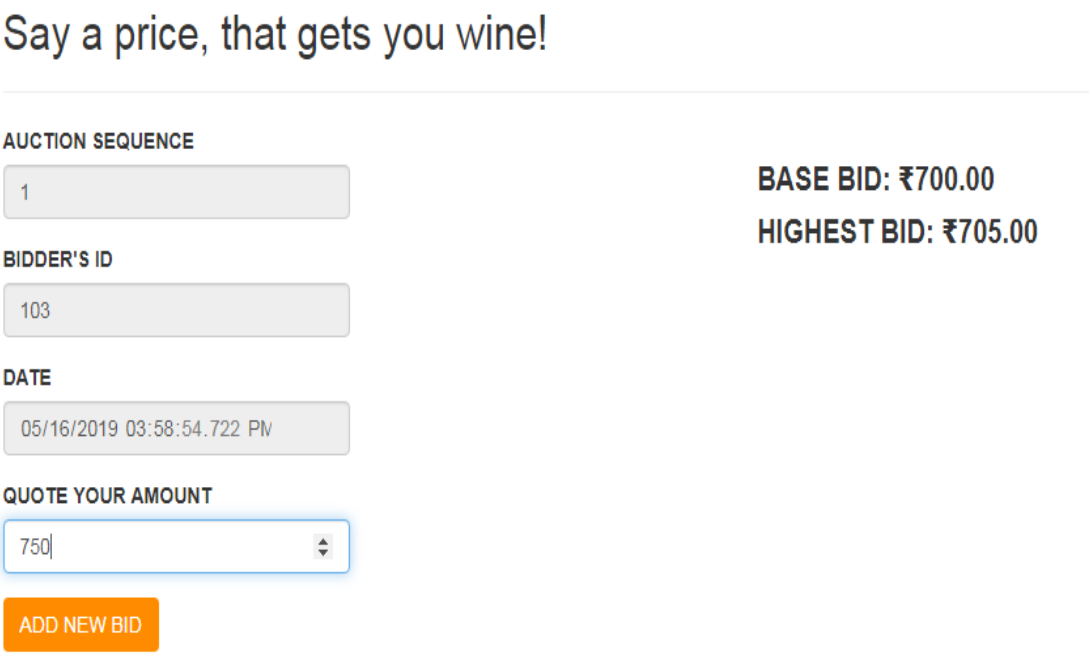


Fig 27. Bid should be more than the Highest Bid Amount

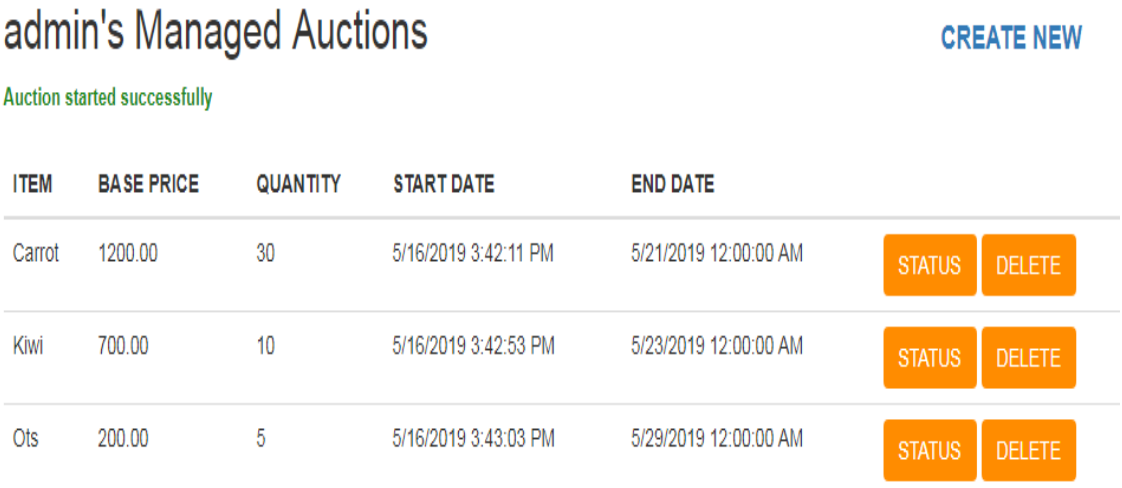


Fig 28. Admin/Employee can check the status (Bids record)

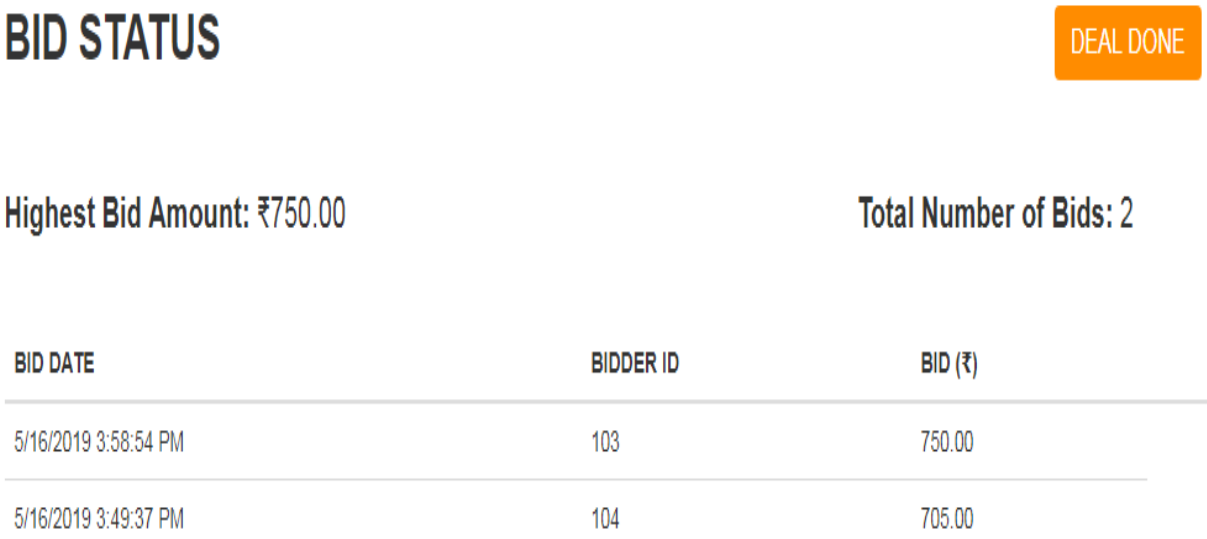


Fig 29. Deal Done by employee will close the Auction and customer with highest bid will be notified

REFERENCES

[1] http://www.w3schools.com/

[2] https://www.tutorialspoint.com/

[3] https://dotnet.microsoft.com/apps/aspnet/mvc

[4] https://crimsonpublishers.com/prsp/pdf/PRSP.000505.pdf

[5] https://www.researchgate.net