MAIN PROJECT

BAKERS LOUNGE

PROJECT REPORT

IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION

SUBMITTED BY:

ANEETA L R

Reg no: 180021096023

UNDER THE GUIDANCE OF

Mrs. MINNU MARTHA SHIBU

2020-2021

DEPARTMENT OF COMPUTER APPLICATION

COCHIN ARTS AND SCIENCE COLLEGE, MANAKKAKADAVU

KAKKANAD

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

PIN: 683565

MAIN PROJECT

BAKERS LOUNGE

PROJECT REPORT

IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION

SUBMITTED BY:

ANEETA L R

Reg no: 180021096023

UNDER THE GUIDANCE OF

Mrs. MINNU MARTHA SHIBU

2020-2021

DEPARTMENT OF COMPUTER APPLICATION

COCHIN ARTS AND SCIENCE COLLEGE, MANAKKAKADAVU

KAKKANAD

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

PIN: 683565

COCHIN ARTS AND SCIENCE COLLEGE, MANAKKAKADAVU

KAKKANAD

(AFFILIATED TO MAHATMA GANDHI UNIVERSITY)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**BAKERS LOUNGE**" submitted to **MAHATMA GANDHI UNIVERSITY** in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Computer Application** is a record of the original work done by **ANEETA L R***(reg no: 180021096023* under my supervision and guidance and that this project work has not formed the basis for the award of any Degree/Diploma/Fellowship or similar title to any candidates of any University.

INTERNAL GUIDE **COLLEGE SEAL** **H.O. D**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SUBMITTED FOR THE UNIVERSITY VIVA-VOICE EXAMINATION HELD ON …….....

DECLARATION

We hereby declare that the mini project work entitled "**BAKERS LOUNGE**" submitted in the partial fulfilment of the requirement for the award of **Bachelor of Computer Application** of **M.G University**, Kottayam is a report of original work done by us during the period of study at **COCHIN ARTS AND SCIENCE COLLEGE,** Manakakkadavu under the supervision and guidance of **Mrs. MINNU MARTHA SHIBU**, Department of Computer Application

Place: **ANEETA L R**
Date:

ACKNOWLEDGEMENT

First of all, We thank the GOD Almighty, for showering his grace upon us to successfully carry out this work, everything in time.

We express our heartfelt gratitude to **Dr. N. Jayakumar**, Principal, Cochin Arts and Science College.

We take this opportunity to thank **Mrs. Sandra Jose**, the Head of the Dept. of Computer Applications, for providing us the necessary facilities to do this work.

We are deeply indebted to **Mrs. Minnu Martha** Shibu, Dept. of Computer Applications, for the excellent guidance and timely suggestions.

Finally, We express our deepest gratitude to all our family members for their encouragement, which helped us to keep our spirit alive and complete this work successfully.

ANEETA L R

ABSTRACT

Bakers lounge is a website mainly for encouraging the people who are good at baking and those who thinking of starting to sell bakery products. This website also helps people to buy bakery items according to their needs without even leaving their house. In this website the admin can view and respond to users, feedback and complaints. The users in this site are bakers, shops, customers and delivery boys. They can register, login, view their own profile and also they can edit their details. Admin can manage login and registration of shops and delivery boys. Bakers can upload their product and cake details and images into the site. Meanwhile customers can book the product they want to buy. They can even search bakers according to location and products needs. The customers can request customized design cakes. Bakers can view the orders and request by customers.

If the baker's are looking for ingredients and tools for baking they can buy them here in this website itself. The shop which sells baking ingredients and tools are also a user in this website. They can only continue to be user if they make a payment. These shops can add ingredients and tools along with its details to sell. These ingredients and tools can be bought by bakers. After booking ingredients and tools they are delivered by the delivery boys. Delivery boys not only deliver ingredients and tools they will also help baker delivery the products and cakes booked by customers. The delivery boys can see their assigned works by both bakers and shops. They can also update the delivery status.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 INTRODUCTION

The project entitled "BAKERS LOUNGE" is a well prepared project in ASP.NET(Visual Studio 2012) and SQL 2014 platform .This software is provided individual bakers to sell their products and buy baking essentials.This website also helps people to buy bakery items according to their needs without even leaving their house. In this website the admin handle values in forms or design and verify shop and delivery boys registering in this site. He can also read and respond to complaints. He can also view other user's details.

The users in this site are bakers, shops, customers and delivery boys. They can register, login, view their own profile and also they can edit their details. Bakers can upload their product and cake details and images into the site. Meanwhile customers can book the product they want to buy from the products bakers uploaded. They can even search bakers, cakes and products according to location. The customers can request customized cakes. Bakers can view the orders and request by customers and assign delivery boy after payment.

If the bakers are looking for ingredients and tools for baking, they can buy them here in this website itself. The shop which sells baking ingredients and tools are also a user in this website. These shops can add ingredients and tools along with its details to sell. These ingredients and tools can be bought by bakers. After booking ingredients and tools they are delivered by the delivery boys assigned by shops. The delivery boys can see their assigned works by both bakers and shops and update delivery status.

.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION TO SYSYTEM ANALYSIS

System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. The outputs from the organization are traced through the various processing that the input phases through in the organization. This involves gathering information and using structured tools for analysis. A detailed study of this process must be made by various techniques like interviews; questionnaires etc.

It is necessary to have such a good system analysis and then by a project development cycle so that the project can be completed in a strictly manner and able to finish with the desired time. The analyst must be so careful about his responsibilities.

As the next step the current system analysis is done which identifies the real need of establishing our project in the environment, its opportunities and constraints etc… All of the steps discussed above are collectively known as the system analysis.

2.2. EXISTING SYSTEM

There are many websites that help people to buy bakery items. These websites help shops(bakery) to grow their income. These websites focus on shops rather than bakers

2.2.1 LIMITATION OF EXISTING SYSTEM

- Doesn’t help individual bakers to start or grow their business
- Some shops don’t make customized cakes

2.3 PROPOSED SYSTEM

The main objective of proposed system is to help customer book bakery items. These bookings are given to individual bakers, the customer can book item by seeing item or by finding their suitable location. The bakers can buy their baking ingredients and tools from the shops registered in this site. The delivery boys registered in this site are assigned to delivery baker’s and customer’s order. The main focus is help individual bakers to earn money.

2.3.1. ADVANTAGES OF PROPOSED SYSTEM

- Help both individual bakers and shops selling products used in baking to earn more money
- By providing tools and ingredients baking becomes easier for home bakers
- Help customer order customized products
- Find shops and bakers according to location and book products
- Provide jobs to people who are willing to work as delivery boy
- Delivery team will help bakers and shops to make their order's delivery easy
- Faster access information
- Efficient Traceability
- Taking into the speed of computer access large data in less time and facilities provided by access.
- Provide rating option for scaling companies
- Better communication option

3. FEASIBILITY ANALYSIS

3.1. INTRODUCTION TO FEASIBILITY ANALYSIS

One of the important outcomes of the preliminary investigation is the determination of the feasibility of the system. These are different aspects of the feasibility study in the investigation phase. After the documents reviewing the selected personnel, investigating the various resources the following are the three feasibility.

Three key combinations are involved in the feasibility study

- Technical feasibility
- Economic feasibility
- Operational feasibility

3.2 TECHNICAL FEASIBILITY

Technical study is a study of hardware and software requirements. All the technical issue related to the proposed system is dealt during feasibility stage of preliminary investigation produced the following results.

3.3 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefit and saving that are expected from a candidate system and compare them with the term of time by automating the process of report generation. The system can be developed technically and if installed would still be good for the organization. The cost is found to be lesser compared to the benefits of the proposed system. The workload of a user will decrease to half of the current workload. Hence the proposed system is found to be economic feasible.

3.4 OPERATIONAL FEASIBILITY

The developed system is completely driven and user friendly. Also, the system is developed in Visual Basic, which is GUI. There is little need skill for new user to operate the software. Reports will be exactly as per the requirement. At the beginning of preliminary investigation work all the personnel approached responded positively this reduces the chance of resistance to the proposed system. Considering all the issue stated above makes the proposed system feasible.

4. SYSTEM SPECIFICATION

4.1. HARDWARE REQUIREMENTS

Selection of hardware configuration is very important task related to the website development. The processor should be powerful to handle all the operations. The hard disk should have the sufficient capacity to solve the database and the application.

Minimum hardware requirements

Processor	:	i3 (Minimum require dual core)
RAM	:	8GB (Minimum required2GB)
HDD	:	1TB free space (Minimum require 500mb)
Monitor	:	DELL/64 Bit LED
CD-R/W Drive	:	52X CD/RW
Printer	:	Standard

4.2. SOFTWARE REQUIREMENTS

Operating System : Windows 10

Front End : Microsoft ASP.NET (2012)

ASP.NET is an addition to VB products. It allows Window application and easy learning which enables us to create programs quickly for PC without being an expert in C++ or other programming language.

Back End : SQL Server 2014

SQL Server 2014 is an essay to use, lightweight version of SQL Server 2014 designed for quickly building data driven applications. The main advantages are,

- Easy to install
- Easy to install and manage
- Rich database functionality

- Studio environment is for great creating almost any type of locations we can think of. We can develop robust stand-alone applications, games and utilities is less time than in order languages. We can also use Active X technology to create internet enabled application that is limited only by our imagination.

4.3 LANGUAGE AND TOOLS USED

4.3.1 FRONT END

MS ASP.NET is the newest of the popular programming language. With its new features, Visual Basic is even stronger contender in the web development areas than ever before.

The Visual ASP.NET is an addition to VB products. It allows Windows application and easy learning which enables us to create programs quickly for PC without being an expert in C++ or C# other programming language

Visual Basic provides a graphical environment where we usually design the forms and control that our website uses. Visual Basic support many useful tools that are more productive. This includes but is not limited to project, forms, objects templates. Custom controls, add-ins and data manager. We can use these tools together to create website in months, weeks, or every day, when compared too much longer development time when we use other languages. Version 6 of Visual Basic is specifically designed to utilize the Internet. It becomes with several controls that allows to create web-based application called Active-executables.

4.3.2 BACK END

SQL Server 2014 is the native data store of C# .NET every business enterprise maintains large volumes of data for its operations. With more and more people accessing data for their work, the need to maintain its integrity and relevance increases. Normally with the traditional method of storing data and information in the files, the chances that the data loses, its integrity and validity are very high. SQL Server 2014 offers capabilities of both relational and object-oriented database systems. In general, objects can be defined as reusable software calls which can be location independent and perform a specific task on any application environment with little or no chance to the code.

SQL Server products are based on a concept known as ‘Client Server Technology’. This concept invokes segregating the processing of an application between two systems. One performs all activities related to database (server) and other performs activities that help the user to interact with the application client.

A client of front-end database applications also interacts with the database by requesting and receiving information from the database server. It acts as interface between the user and the database. The Database Server or back-end is used to manage the database tables optimally among multiple clients who concurrently request the server for same data. It also enforces data integrity across all client application and controls database access and other.

Microsoft SQL Server 2014 extends the performance, reliability, quality and ease-of-use of Microsoft SQL Server version 8.00194 Microsoft SQL Server 2014 includes several new features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce applications.

The OLAP Services feature available in SQL Server version 8.0 is now called SQL Server 2014 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component.

The Repository component available in SQL Server version 8.0 is now called Microsoft SQL Server 2014 Meta Data Services. References to the component now use the term Meta Data Services. The term Repository is used only in reference to the repository engine with in Meta Data Services.

4.3.3 OPERATING SYSTEM

The OS used is Windows Operating System. The hall mark software of Microsoft, which had created a new wave of graphical user interface in the industry, WINDOWS 7 and 10 stands in the top of its popularity. The advent of Microsoft plus has cured whatever faults were there in the original WINDOWS 10 version and made it and useful tool to work with the memory resident programs of it, make the reloading of WINDOWS 10 easier, it plugs and play connectivity for input output devices makes a new dimension towards the use of computer system. Connectivity to the information networks like Internet through modems makes it overstate software. Almost all new software has their windows version also. The programmer and file manager facilities of it had made a leap way towards giving a new dimension towards the operation of computer systems

5. DATA FLOW DIAGRAM

5.1 INTRODUCTION TO DATA FLOW DIAGRAM

To start the system design, something analogue to the architecture blue print as a starting point to design is required. It is a way to focus on functions rather than physical implementation. One such tool is a DFD.

Structured analysis is a set of techniques and graphical tools that help the analyst to develop a new kind of system specification that are easily understandable to the user. DFD's show the major decompositions of the system functions and their interfaces. The DFD is graphic and presents a picture of what is being specified and is conceptually easy to understand presentation of the application.

One important feature of DFD's is that it is logical rather than physical. The elements of the system do not depend on vendor or hardware. They specify in precise, concise manner the working of the system and how it hangs together.

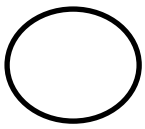
DFD is the graphic representation of data movement process, and files used in support of an information system. There are several rules of thumb used in drawing DFDs.

- Process should be named and numbered for easy references.
- The direction of flow is from top to bottom and from left to right.
- When a process is imported in the lower level's details, they must be numbered.
- Process and data flow names have the first letter of each word must be a capital letter.

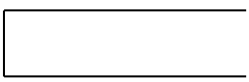
5.2 BASIC DATA FLOW DIAGRAM SYMBOLS

Source or destination of data

Data flow



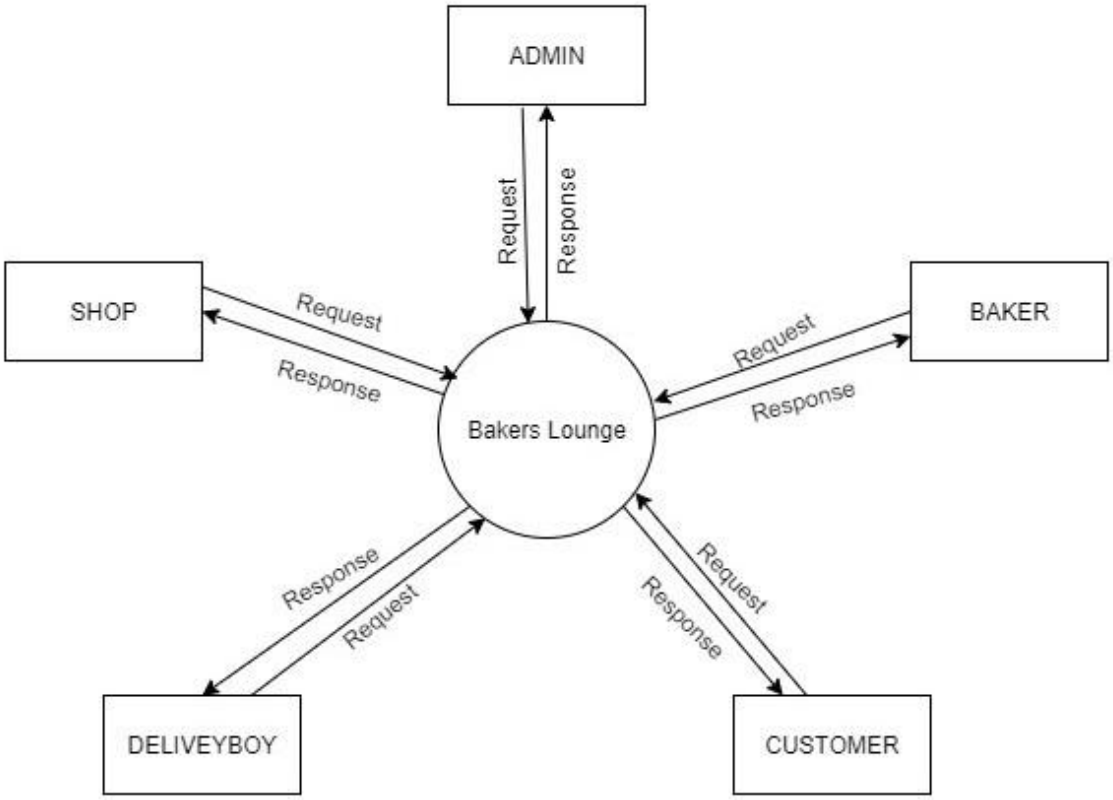
Process that transforms data flow



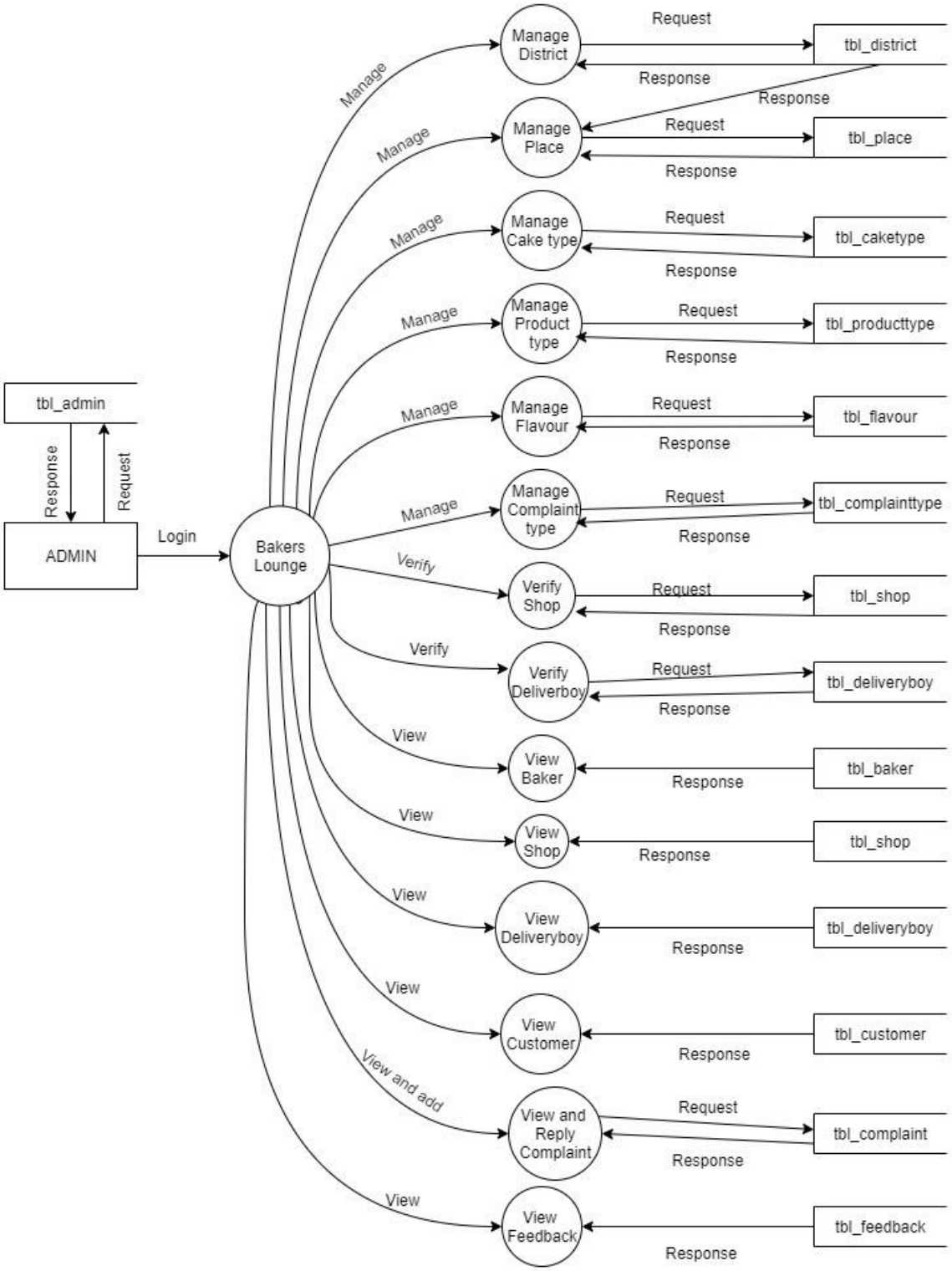
Data store

5.3 DATA FLOW DIAGRAM SHOWING PROPOSED SYSTEM

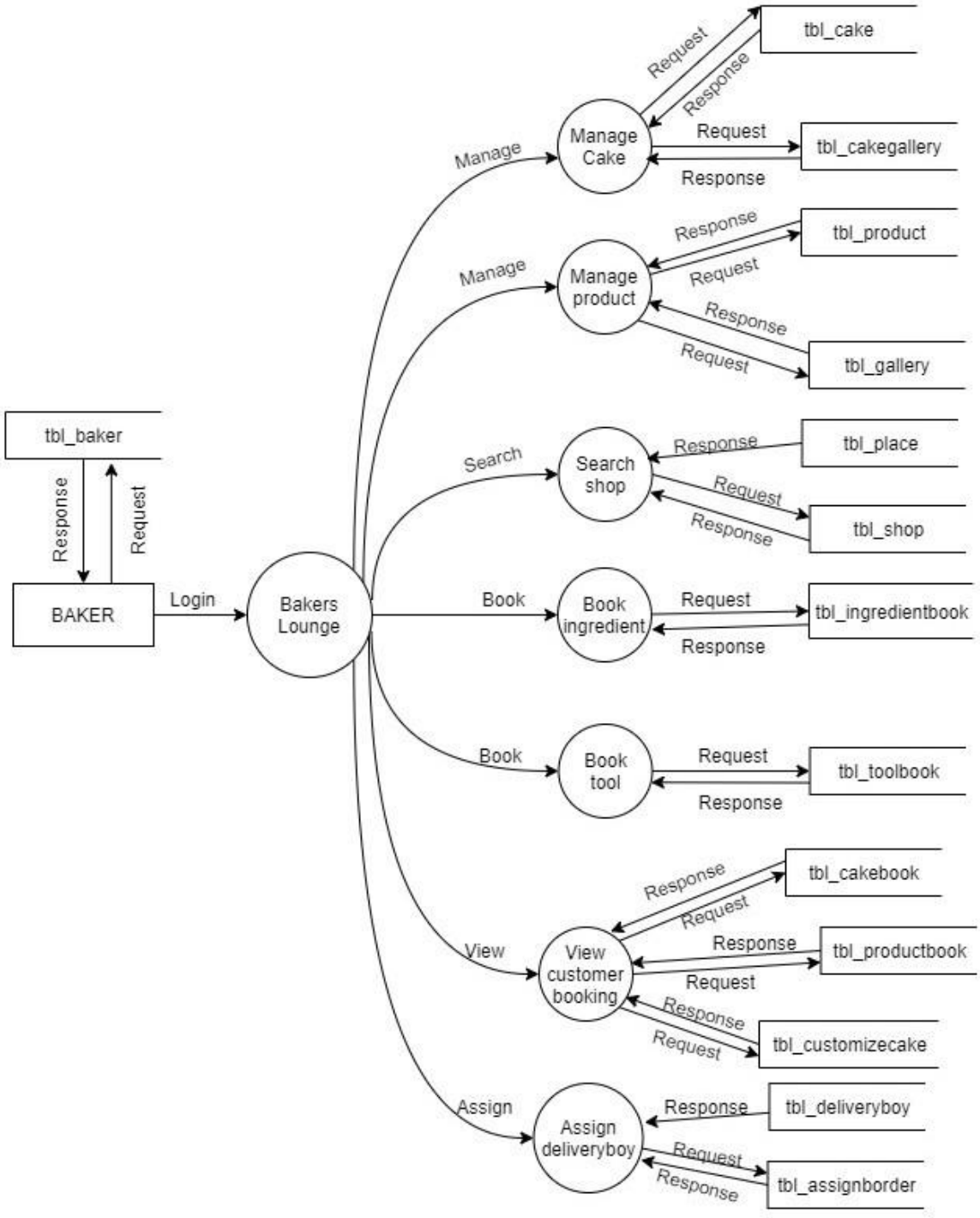
Level 0 – System



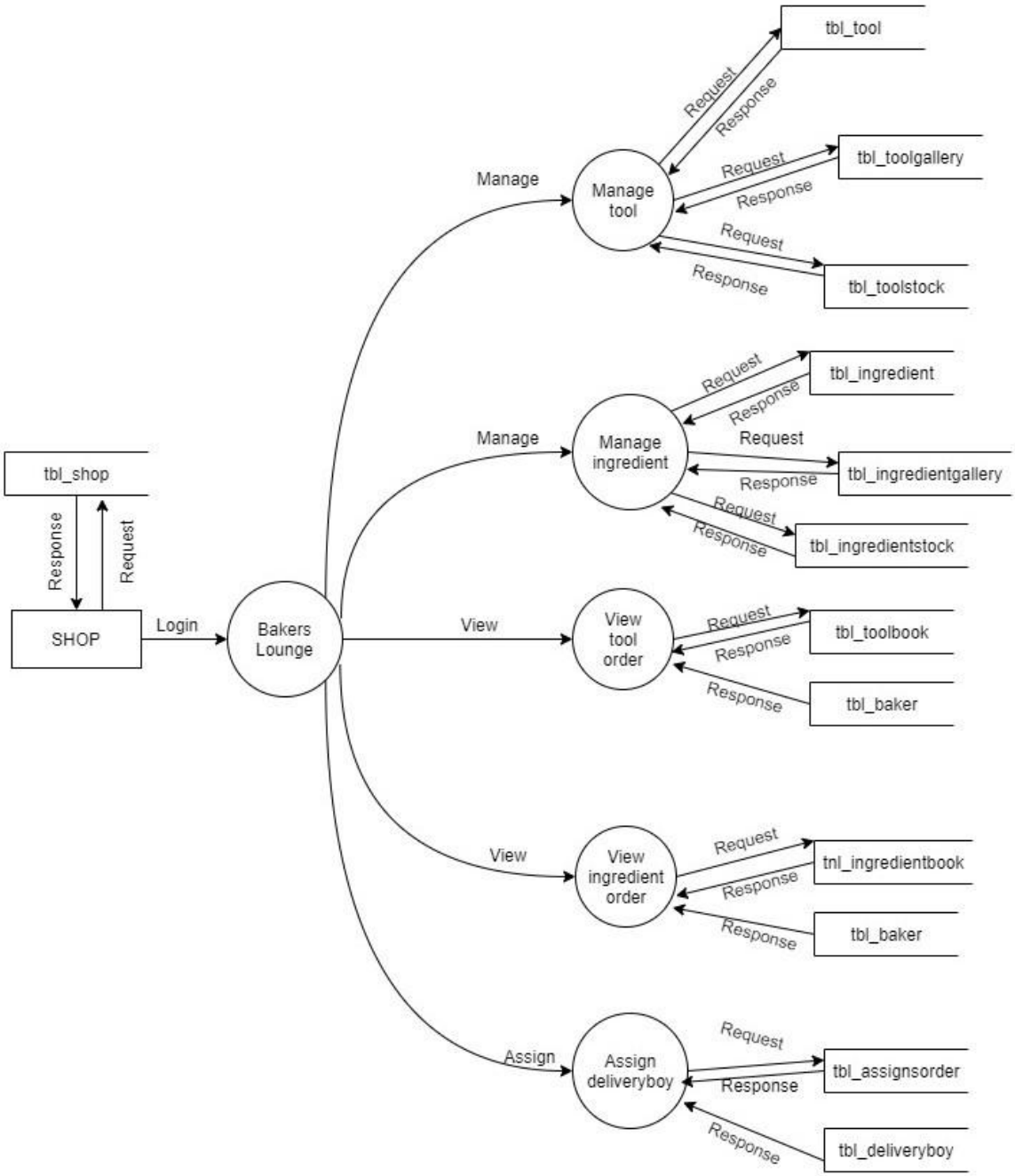
Level 1-Admin



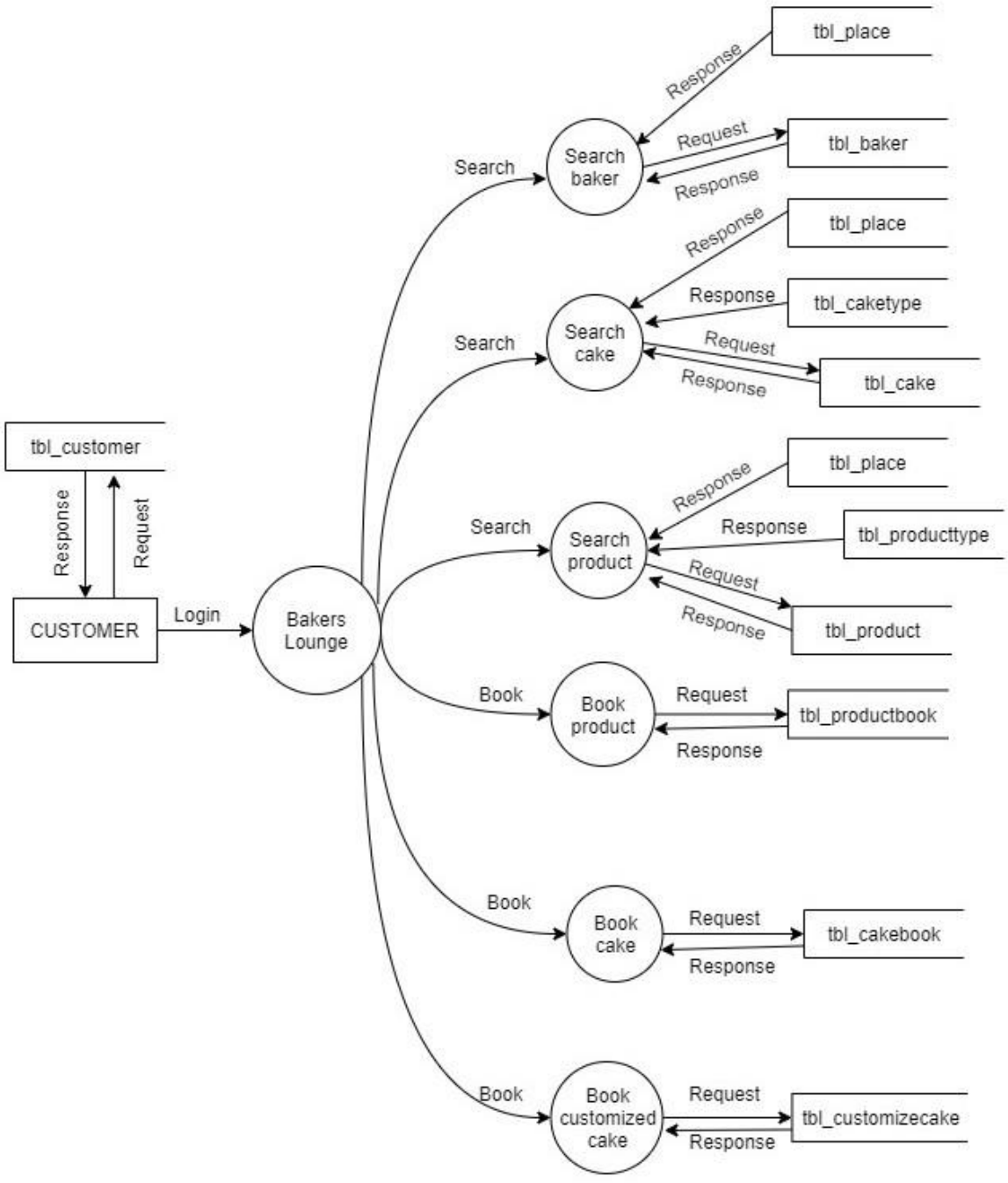
Level 1- Baker



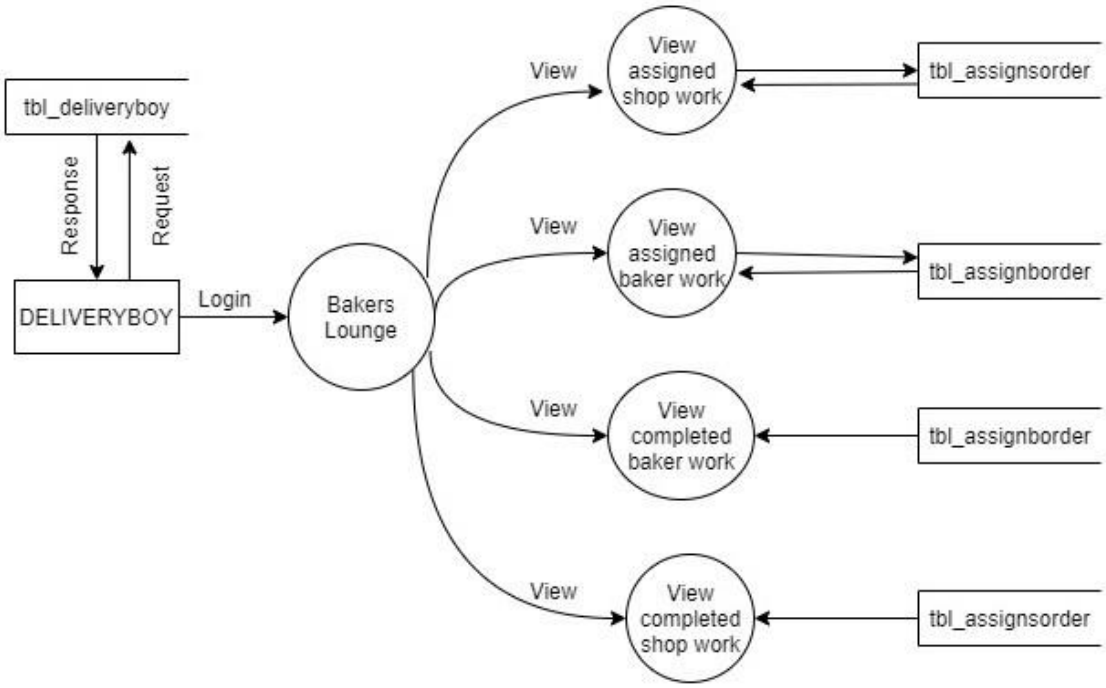
Level 1- Shop


tbl_shop
Response
Request
SHOP
Login
Bakers Lounge
Manage
Manage tool
tbl_tool
tbl_toolgallery
tbl_toolstock
Manage
Manage ingredient
tbl_ingredient
tbl_ingredientgallery
tbl_ingredientstock
View
View tool order
tbl_toolbook
tbl_baker
View
View ingredient order
tnl_ingredientbook
tbl_baker
Assign
Assign deliveryboy
tbl_assignsorder
tbl_deliveryboy
Request
Response

Level 1- Customer



Level 1- Deliveryboy



6. SYSTEM DESIGN

6.1 INTRODUCTION TO SYSTEM DESIGN

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used.

The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

6.2 LOGICAL DESIGN

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes , frequencies etc.
- Prepares output specifications – that is, determines the format, content and frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

6.3 PHYSICAL DESIGN

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design Walk through.
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits , costs , conversion date and system constraints

Design/Specification activities:

- Concept formulation.
- Problem understanding.
- High level requirements proposals.
- Feasibility study.
- Requirement's engineering.
- Architectural design.

6.4 INPUT DESIGN

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

o What data should be given as input?

o How the data should be arranged or coded?

o The dialog to guide the operating personnel in providing input.

o Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in a maze of instant. Thus, the objective of input design is to create an input layout that is easy to follow

6.5 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should:

ϖ Identify the specific output that is needed to meet the requirements.

ϖ Select methods for presenting information.

ϖ Create document, report, or other formats that contain information produced by the system.

6.6 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected. The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used.

A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

ϖ Data Integrity

ϖ Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- To structure the data so that there is no repetition of data , this helps in saving.
- To permit simple retrieval of data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions, deletions.
- To reduce the need to restructure or reorganize data which new application requirements arise

.6.6.1 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

RELATIONS, DOMAINS & ATTRIBUTES:

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

6.6.2 RELATIONSHIPS:

Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys. Entity Integrity enforces that no Primary Key can have null values. Referential Integrity enforces that no Primary Key can have null values.

Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

6.6.3 NORMALIZATION:

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources.

These includes:

ϖ Normalize the data.

ϖ Choose proper names for the tables and columns.

ϖ Choose the proper name for the data.

First Normal Form:

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible value.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form:

According to Second Normal Form, For relations where primary key contains

multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key .

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form:

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key.
In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

6.7 TABLE DESIGN

Table Name: tbl_admin

Table Description: Details about admin

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	admin_id	int	4	Primary key
2.	admin_name	nvarchar	50	
3.	admin_password	nvarchar	50	

Table Name: tbl_district

Table Description: District added by admin

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	district_id	int	4	Primary key
2.	district_name	nvarchar	50	

Table Name: tbl_place

Table Description: Places in district added by admin

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	place_id	int	4	Primary key
2.	place_name	nvarchar	50	
3.	district_id	int	4	Foreign key

Table Name: tbl_caketype

Table Description: Cake type to differentiate cakes while uploading and searching

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	caketype_id	int	4	Primary key
2.	caketype_name	nvarchar	50	

Table Name: tbl_producttype

Table Description: Product type to differentiate cakes while uploading and searching

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	producttype_id	int	4	Primary key
2.	producttype_name	nvarchar	50	

Table Name: tbl_deliveryboy

Table Description: Details about deliveryboys

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	deliveryboy_id	int	4	Primary key
2.	deliveryboy_name	nvarchar	50	
3.	deliveryboy_gender	nvarchar	50	
4.	deliveryboy_address	nvarchar	50	
5.	deliveryboy_proof1	nvarchar	50	
6.	deliveryboy_proof2	nvarchar	50	
7.	deliveryboy_photo	nvarchar	50	
8.	deliveryboy_contact	nvarchar	50	
9.	deliveryboy_email	nvarchar	50	
10.	deliveryboy_username	nvarchar	50	
11.	deliveryboy_password	nvarchar	50	
12.	deliveryboy _status	nvarchar	50	
13.	place_id	int	4	Foreign key

Table Name: tbl_baker

Table Description: Details about bakers

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	baker_id	int	4	Primary key
2.	baker_name	nvarchar	50	
3.	baker_gender	nvarchar	50	
4.	baker_address	nvarchar	50	
5.	baker_proof	nvarchar	50	
6.	baker_photo	nvarchar	50	
7.	baker_contact	nvarchar	50	
8.	baker_email	nvarchar	50	
9.	baker_username	nvarchar	50	
10.	baker_password	nvarchar	50	
11.	baker_status	nvarchar	50	
12.	place_id	int	4	Foreign key

Table Name: tbl_customer

Table Description: Details about customers

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	user_id	int	4	Primary key
2.	user_name	nvarchar	50	
3.	user_address	nvarchar	50	
4.	user_contact	nvarchar	50	
5.	user_email	nvarchar	50	
6.	user_username	nvarchar	50	
7.	user_password	nvarchar	50	
8.	place_id	int	4	Foreign key

Table Name: tbl_toolgallery

Table Description: Address or path of images of tools

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	toolgallery_id	int	4	Primary key
2.	toolgallery_image	nvarchar	50	
3.	tool_id	int	4	Foreign key

Table Name: tbl_shop

Table Description: Details about shops

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	shop_id	int	4	Primary key
2.	shop_name	nvarchar	50	
3.	shop _logo	nvarchar	50	
4.	shop _address	nvarchar	50	
5.	shop _proof	nvarchar	50	
6.	baker_licno	nvarchar	50	
7.	shop _contact	nvarchar	50	
8.	shop _email	nvarchar	50	
9.	shop_ownername	nvarchar	50	
10.	shop_ownercontact	nvarchar	50	
11.	shop_owneraddr	nvarchar	50	
12.	shop_ownerproof	nvarchar	50	
13.	shop _username	nvarchar	50	
14.	shop _password	nvarchar	50	
15.	shop _status	nvarchar	50	
16.	Shop_pstatus	nvarchar	50	
17.	place_id	int	4	Foreign key

Table Name: tbl_tool

Table Description: Details about tools

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	tool_id	int	4	Primary key
2.	tool_name	nvarchar	50	
3.	tool_description	nvarchar	50	
4.	tool_price	nvarchar	50	
5.	tool_image	nvarchar	50	
6.	shop_id	int	4	Foreign key

Table Name: tbl_ingredient

Table Description: Details about ingredient

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	ingredient_id	int	4	Primary key
2.	ingredient_name	nvarchar	50	
3.	ingredient_description	nvarchar	50	
4.	ingredient_price	nvarchar	50	
5.	Ingredient_expiry	nvarchar	50	
6.	ingredient _image	nvarchar	50	
7.	shop_id	int	4	Foreign key

Table Name: tbl_ingredientgallery

Table Description: Address or path of images of ingredient

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	ingallery_id	int	4	Primary key
2.	ingallery_image	nvarchar	50	
3.	ingredient_id	int	4	Foreign key

Table Name: tbl_cake

Table Description: Details about cakes

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	cake_id	int	4	Primary key
2.	cake _name	nvarchar	50	
3.	cake _description	nvarchar	50	
4.	cake _price	nvarchar	50	
5.	cake _weight	nvarchar	50	
6.	cake _image	nvarchar	50	
7.	caketype_id	int	4	Foreign key
8.	baker_id	int	4	Foreign key

Table Name: tbl_toolstock

Table Description: Details about tool stock

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	toolstock_id	int	4	Primary key
2.	toolstock_quantity	nvarchar	50	
3.	tool_id	int	4	Foreign key

Table Name: tbl_ingredientstock

Table Description: Details about ingredient stock

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	instock_id	int	4	Primary key
2.	instock_image	nvarchar	50	
3.	instock_id	int	4	Foreign key

Table Name: tbl_gallery

Table Description: Address or path of images of bakery products

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	gallery_id	int	4	Primary key
2.	gallery_image	nvarchar	50	
3.	cake_id	int	4	Foreign key

Table Name: tbl_cakegallery

Table Description: Address or path of images of cakes

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	cakegallery_id	int	4	Primary key
2.	cakegallery_image	nvarchar	50	
3.	cake_id	int	4	Foreign key

Table Name: tbl_product

Table Description: Details about bakery product

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	product_id	int	4	Primary key
2.	product_name	nvarchar	50	
3.	product_description	nvarchar	50	
4.	product_price	nvarchar	50	
5.	product_quantity	nvarchar	50	
6.	product _image	nvarchar	50	
7.	producttype_id	int	4	Foreign key
8.	baker_id	int	4	Foreign key

Table Name: tbl_complaint

Table Description: Complaints added by users,bakers and shops

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	complaint_id	int	4	Primary key
2.	complainttype_id	int	4	Foreign key
3.	complaint_title	nvarchar	50	
4.	complaint_data	nvarchar	50	
5.	complaint_date	nvarchar	50	
6.	complaint_reply	nvarchar	50	
7.	complaint_replydate	nvarchar	50	
8.	complaint_status	nvarchar	50	
9.	user_id	int	4	Foreign key
10.	baker_id	int	4	Foreign key
11.	shop_id	int	4	Foreign key

Table Name: tbl_assignsorder

Table Description: Details of assigned work to delivery boy by shops

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	assignsorder_id	int	4	Primary key
2.	assignsorder_date	nvarchar	50	
3.	assignsorder_status	nvarchar	50	
4.	tool_id	int	4	Foreign key
5.	ingredient_id	int	4	Foreign key
6.	deliveryboy_id	int	4	Foreign key

Table Name: tbl_toolbook

Table Description: Details about tool bookings

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	toolbook_id	int	4	Primary key
2.	toolbook_quantity	nvarchar	50	
3.	toolbook_totalamount	nvarchar	50	
4.	toolbook_date	nvarchar	50	
5.	toolbook_deliveryaddr	nvarchar	50	
6.	toolbook_deliverydate	nvarchar	50	
7.	toolbook_cstatus	nvarchar	50	
8.	toolbook_pstatus	nvarchar	50	
9.	toolbook_dstatus	nvarchar	50	
10.	tool_id	int	4	Foreign key
11.	baker_id	int	4	Foreign key

Table Name: tbl_ingredientbook

Table Description: Details about ingredient bookings

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	inbook_id	int	4	Primary key
2.	inbook_quantity	nvarchar	50	
3.	inbook_totalamount	nvarchar	50	
4.	inbook_date	nvarchar	50	
5.	inbook_deliveryaddr	nvarchar	50	
6.	inbook_deliverydate	nvarchar	50	
7.	inbook_cstatus	nvarchar	50	
8.	inbook_pstatus	nvarchar	50	
9.	inbook_dstatus	nvarchar	50	
10.	ingredient_id	int	4	Foreign key
11.	baker_id	int	4	Foreign key

Table Name: tbl_cakebook

Table Description: Details about cake bookings

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	cakebook_id	int	4	Primary key
2.	cakebook_quantity	nvarchar	50	
3.	cakebook_totalamount	nvarchar	50	
4.	cakebook_date	nvarchar	50	
5.	cakebook_deliveryaddr	nvarchar	50	
6.	cakebook_deliverydate	nvarchar	50	
7.	cakebook_caption	nvarchar	50	
8.	cakebook_pickupaddr	nvarchar	50	
9.	cakebook_cstatus	nvarchar	50	
10.	cakebook_pstatus	nvarchar	50	
11.	cakebook_dstatus	nvarchar	50	
12.	cake_id	int	4	Foreign key
11.	user_id	int	4	Foreign key

Table Name: tbl_assignborder

Table Description: Details of assigned work to delivery boy by bakers

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	assignborder_id	int	4	Primary key
2.	assignborder_date	nvarchar	50	
3.	assignborder_status	nvarchar	50	
4.	ccake_id	int	4	Foreign key
5.	cake_id	int	4	Foreign key
6.	product_id	int	4	Foreign key
7.	deliveryboy_id	int	4	Foreign key

Table Name: tbl_complainttype

Table Description: Complainttype value is used to differentiate complaints

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	complainttype_id	int	4	Primary key
2.	complainttype_name	nvarchar	50	

Table Name: tbl_customizecake

Table Description: Details about customized cake bookings

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	ccake_id	int	4	Primary key
2.	ccake _shape	nvarchar	50	
3.	ccake _caption	nvarchar	50	
4.	ccake _detail	nvarchar	50	
5.	ccake _weight	nvarchar	50	
6.	ccake _date	nvarchar	50	
7.	ccake_deliveryaddr	nvarchar	50	
8.	ccake _deliverydate	nvarchar	50	
9.	ccake _pickupaddr	nvarchar	50	
10.	ccake _status	nvarchar	50	
11.	ccake _pstatus	nvarchar	50	
12.	ccake _dstatus	nvarchar	50	
13.	ccake_totalamount	nvarchar	50	
14.	flavour_id	Int		
15.	user_id	int	4	Foreign key
16	baker_id	int	4	Foreign key

Table Name: tbl_productbook

Table Description: Details about product bookings

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	productbook_id	int	4	Primary key
2.	productbook_quantity	nvarchar	50	
3.	productbook_totalamount	nvarchar	50	
4.	productbook_date	nvarchar	50	
5.	productbook_deliveryaddr	nvarchar	50	
6.	productbook_deliverydate	nvarchar	50	
7.	productbook_pickupaddr	nvarchar	50	
8.	productbook_cstatus	nvarchar	50	
9.	productbook_pstatus	nvarchar	50	
10.	productbook_dstatus	nvarchar	50	
11.	product_id	int	4	Foreign key
12.	user_id	int	4	Foreign key

Table Name: tbl_flavour

Table Description: Values used in customized cake form

SL NO.	FIELD	TYPE	SIZE	COMMENT
1.	flavour_id	int	4	Primary key
2.	flavour_name	nvarchar	50	

7. SYSTEM TESTING

7.1 INTRODUCTION TO SYSTEM TESTING

Testing is the process of examining the software to compare the actual behavior with that of the excepted behavior. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

The testing steps are,

- Unit testing
- Integration testing
- Validation testing
- Output Testing
- User Acceptance Testing
- Alpha Testing
- Beta Testing

System Testing provides the file assurance that software once validated must be combined with all other system elements. System testing verifies whether all elements have been combined properly and that overall system function and performance is achieved. After the integration of modules, the validation test was carried out over the system. It was found that all the modules work well together and meet the overall system function and performance.

7.2 UNIT TESTING

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Various test cases are prepared. For each module these test cases are implemented and it is checked whether the module is executed as per the requirements and outputs the desired result. In this test each service input and output parameters are checked.

In unit testing,

- Module interface was tested to ensure that information properly flows into and out of the program under test.
- Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
- All independent paths through the control structures were executed to ensure that all statements in the modules have been executed at least once.

Error handling paths were also tested.

7.3 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here.

The modules are tested separately for accuracy and modules are integrated together using bottom up integration i.e., by integrating from moving from bottom to the top. The system is checked and errors found during integration are rectified. In this testing, all the individual modules were combined and the module wise shifting was verified to be alright.

The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis, and display processing and database management.

The integration testing is performed in the “Tech Mate” by combining the three modules i.e.; by combining the technician and user modules, or by admin and technician modules or by combining the admin and technician modules and found all are running without any error.

7.4 VALIDATION TESTING

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the customer under validation is alpha and beta testing. Alpha testing is where the end user tests the system rather than the developer, but in a controlled environment. The software is used on a natural setting with the developer monitoring the user using the system. The developer records the errors and usage problems encountered by the user.

7.5 OUTPUT TESTING

After performing the validation testing ,the next step is output testing of the proposed system, since could be useful if it does not produce the required output in the specific format. The output generated by the system under considerations is tested asking the users about the format required by them.

7.6 USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under considerations is tested for users acceptance by constantly keeping in touch with the prospective system users at the time of developing and making of hardware management software.

7.7 ALPHA TESTING

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

7.8 BETA TESTING

Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

The user person conducts beta testing at one more sites. The developer is not present during these tests. Hence, beta test can be said as the live application of software on an environment that cannot be controlled by the developer. The sales person takes down the problems encountered during beta testing and reports to the developer at regular intervals. The developer makes suitable modifications to the software henceforth.

The client-side validation is done in this testing phase. It is checked whether the data passed to each service is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

8. IMPLEMENTATION

8.1 IMPLEMENTATION

The implementation is the final state and it is an important phase. It involves the individual programming; system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users, which should really have been taken place much earlier in the project when they were being involved in the investigation and design work. During the implementation phase system actually takes physical shape. In order to develop a system implemented planning is very essential.

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented. Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right. The implementation stage involves following tasks:

IMPLEMENTATION PLAN

- Test system with sample data
- Detection and correction of errors
- Make the necessary changes in the system
- Check with the existing system
- Installation of hardware and software utilities
- Training and involvement of user personnel

9. MAINTENANCE

9.1 MAINTENANCE

Maintenance is actually the implementation of the post implementation review plan. As important as it is, many programmers and analyst are reluctant to perform or identify themselves with maintains effort. There are psychological, personalities and professional reasons for this. In any case first class effort must be made to ensure that the software changes are made properly and in time to keep the system intone with user specification. Maintenance is expensive. One way to reduce Maintenance costs are through maintenance management and software modifications audits. Software modifications consists of program rewrites system level updates, re-audits of low ranking software, reduced maintenance backlog and higher satisfaction and moral among the maintenance of staff.

Even when the new system has gone live there may be need for some system design activity. This will see from changes that are necessitated by the dynamic nature of the system and its environment. Changes may be required to correct faults or to bring about improvements, and may arise as part of normal running of the system or as a result of a review of the system performance. Changes or amendments and documentations and perhaps handled by specialist group of maintenance staff

10. SECURITY MEASURES

10.1 SECURITY MEASURES

A computer system is secure if neither its availability to attain its objectives not its availability to survive can be adversely affected by an unwanted event.A computer based security is the combination of many assets or resourses designed to perform some function or to provide service.The facility to impose strict authorization is completely vested in the hands of the system respectively.Only valid users can enter into the system.They have to provide a valid username and password to show that they are valid users.if any one of them is wrong,access is denied.At register time admin provide a OTP security check up for user account validation and its help to admin avoid fake ids.

System security is refers to the technical innovation and procedures applied to the hardware and operating system to protect against deliberate or accidental damage from a defined threat. In contrast, data security is the protection of data loss, disclosure, modification and destruction. The security features are considered while developing the system, so as to avoid the errors and omissions that may lead to serious problems.

11. FUTURE ENHANCEMENTS

11.1 FUTURE ENHANCEMENTS

In future we can expect the modified version of Bakers Lounge. The system is very flexible for further up gradation with the visual studio 2012 and sql server 2014 make this modifications very easily it is also possible to involve more functions into the system.

In future we can add a page in customers homepage to book a theme came like booking customized cake. The difference between theme cake book and customized cake booking is that customized cake cannot be made in other shapes or flavor that is selected in the form. But in theme cake booking customer can attach a photo showing in which shape or design it should be made and they can ask for mixed flavors or layers of different flavors. Likewise there are many other ideas to make this system more enhanced in future.

12. CONCLUSION

12.1 CONCLUSION

The administrator has the freedom to modify some form values. The commercial deals are done between shops and bakers or customers and baker administrator has no role in it. The bakers or shop are ones that assign delivery boys after the order payment. This system was tested with a lots of test details and found to be working correctly. All the requirements are built into the system. It produces data output in the required format.

The developed system is developed using ASP.NET and Microsoft SQL Server .The system can be easily modified or expanded. The system provides all the facilities required for the bakers, the system is user friendly. The system requires less effort to operate.

Additional functions can be easily added when necessary. The system is developed with modular approach. All modules in the system have been tested and put together to form the main system. Finally the system tested with real data and everything worked successfully. Thus the system has fulfilled all the objectives.

13. REFERENCE

13.1 REFERENCE

[1] Stack Over Flow Web Site

[2] Peter Norton, ”Guide to Visual Studio**”**

[3] Peter Norton ”Software Engineering”

[4] Korth, Sudarshan **,** “Database System Concepts

14. APPENDIX

14.1 CODE

14.1.1 App code

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI.WebControls;

public class bakers
{
    SqlConnection con = new SqlConnection("Data Source=LAPTOP-
D7NBR34I;Initial Catalog=db_bakers;Integrated Security=True");

    public bakers()
    {
        con.Open();
        //
        // TODO: Add constructor logic here
        //
    }
    public void ExecuteCommands(string querystr)
    {
        SqlCommand cmd = new SqlCommand(querystr, con);
        cmd.ExecuteNonQuery();
    }
    public DataTable GetDataTable(string querystr)
    {
        SqlDataAdapter ada = new SqlDataAdapter(querystr, con);
        DataTable dt = new DataTable();
        ada.Fill(dt);
        return dt;
```

```
    }
    public void FillGridView(string querystr, GridView grid)
    {
        DataTable dt = GetDataTable(querystr);
        grid.DataSource = dt;
        grid.DataBind();
    }
    public void FillDataList(string querystr, DataList list)
    {
        DataTable dt = GetDataTable(querystr);
        list.DataSource = dt;
        list.DataBind();
    }
    public void FillDetailsView(string querystr, DetailsView list)
    {
        DataTable dt = GetDataTable(querystr);
        list.DataSource = dt;
        list.DataBind();
    }
    public void FillDrop(DropDownList drop, string display, string
valfield, string tbl)
    {
        string str = "select " + display + "," + valfield + " from "
+ tbl + " order by "+display+" ASC";
        DataTable dt1 = GetDataTable(str);
        drop.DataSource = dt1;
        drop.DataTextField = display;
        drop.DataValueField = valfield;
        drop.DataBind();
        drop.Items.Insert(0, "---select---");

    }
```

```
    public void FillDrop(DropDownList ddl, string display, string
valfield, string tbl, string condition1)
    {
        string str = "select " + display + "," + valfield + " from "
+ tbl + " where " + condition1 + "";
        DataTable dt1 = new DataTable();
        dt1 = GetDataTable(str);
        ddl.DataSource = dt1;
        ddl.DataTextField = display;
        ddl.DataValueField = valfield;
        ddl.DataBind();
        ddl.Items.Insert(0, "---select---");

    }
    public void FillCheck(CheckBoxList chk, string display, string
valfield, string tbl)
    {
        string str = "select " + display + "," + valfield + " from "
+ tbl + "";
        DataTable dt1 = GetDataTable(str);
        chk.DataSource = dt1;
        chk.DataTextField = display;
        chk.DataValueField = valfield;
        chk.DataBind();
    }
    public void FillCheck(CheckBoxList chk, string display, string
valfield, string tbl, string condition1)
    {
        string str = "select " + display + "," + valfield + " from "
+ tbl + " where " + condition1 + "";
        DataTable dt1 = new DataTable(str);
        chk.DataSource = dt1;
        chk.DataTextField = display;
```

```
        chk.DataValueField = valfield;
        chk.DataBind();
    }


    public void FillGridView(string sel, DetailsView dv)
    {
        throw new NotImplementedException();
    }

}
```

14.1.2 Login.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Guest_Default : System.Web.UI.Page
{
    bakers obj = new bakers();
    protected void Page_Load(object sender, EventArgs e)
    {
        Response.Cache.SetCacheability(HttpCacheability.NoCache);
        Response.Cache.SetExpires(DateTime.Now.AddSeconds(-1));
        Response.Cache.SetNoStore();
    }
    protected void btn_login_Click(object sender, EventArgs e)
    {
```

```
        string sel = "select * from tbl_baker where
baker_username='"+txt_uname.Text+"' and
baker_password='"+txt_password.Text+"' and baker_status=1";
        DataTable dt=obj.GetDataTable(sel);

        string sel0 = "select * from tbl_shop where shop_username='"
+ txt_uname.Text + "' and shop_password='" + txt_password.Text + "'
and shop_status=1";
        DataTable dt0 = obj.GetDataTable(sel0);

        string sel1 = "select * from tbl_customer where
user_username='" + txt_uname.Text + "' and user_password='" +
txt_password.Text + "'";
        DataTable dt1 = obj.GetDataTable(sel1);

        string sel2 = "select * from tbl_deliveryboy where
deliveryboy_username='" + txt_uname.Text + "' and
deliveryboy_password='" + txt_password.Text + "' and
deliveryboy_status=1";
        DataTable dt2 = obj.GetDataTable(sel2);

        string sel3 = "select * from tbl_admin where admin_name='" +
txt_uname.Text + "' and admin_password='" + txt_password.Text + "'";
        DataTable dt3 = obj.GetDataTable(sel3);

        if (dt.Rows.Count > 0)
        {
            Session["bid"] = dt.Rows[0]["baker_id"].ToString();
            Response.Redirect("../Baker/BakerHomePage.aspx");
        }
        else if (dt0.Rows.Count > 0)
        {
            Session["sid"] = dt0.Rows[0]["shop_id"].ToString();
```

```
            Response.Redirect("../Shop/ShopHomePage.aspx");
        }
        else if (dt1.Rows.Count > 0)
        {
            Session["cid"] = dt1.Rows[0]["user_id"].ToString();
            Response.Redirect("../Customer/HomePage.aspx");
        }
        else if (dt2.Rows.Count > 0)
        {
            Session["dbid"] =
dt2.Rows[0]["deliveryboy_id"].ToString();
            Response.Redirect("../Deliveryboy/DBHomePage.aspx");
        }
        else if (dt3.Rows.Count > 0)
        {
            Session["aid"] = dt3.Rows[0]["admin_id"].ToString();
            Response.Redirect("../Admin/AdminDashboard.aspx");
        }
    }
}
```

14.1.3 VerifyShop.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Admin_Default : System.Web.UI.Page
{
    bakers obj = new bakers();
```

```
    public static int id;
    protected void Page_Load(object sender, EventArgs e)
    {
        fillgrid();
    }
    public void fillgrid()
    {
        string sel="select * from tbl_shop s inner join tbl_place p
on s.place_id=p.place_id inner join tbl_district d on
p.district_id=d.district_id where shop_status=0 order by shop_id
ASC";
        obj.FillGridView(sel, grd_shop);
        string sel1 = "select * from tbl_shop s inner join tbl_place
p on s.place_id=p.place_id inner join tbl_district d on
p.district_id=d.district_id where shop_status=2 order by shop_id
ASC";
        obj.FillGridView(sel1, grd_reject);
    }
    protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "accept")
        {
            string update = "update tbl_shop set shop_status='1'
where shop_id='" + id + "'";
            obj.ExecuteCommands(update);
            Response.Write("<script>alert('Accepted')</script>");
        }
        if (e.CommandName == "del")
        {
            string update = "update tbl_shop set shop_status='2'
where shop_id='" + id + "'";
```

```
            obj.ExecuteCommands(update);
            Response.Write("<script>alert('Rejected')</script>");
        }
        fillgrid();
    }
    protected void grd_reject_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "retrieve")
        {
            string update = "update tbl_shop set shop_status='0'
where shop_id='" + id + "'";
            obj.ExecuteCommands(update);
            Response.Write("<script>alert('Retrieved')</script>");

        }
    }
}
```

14.1.4 Cake.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.IO;

public partial class Baker_Default : System.Web.UI.Page
{
    bakers obj = new bakers();
```

```
    public static int id;
    protected void Page_Load(object sender, EventArgs e)
    {
        fillgrid();
        if (!IsPostBack)
        {
            filldrop();
        }
    }
    protected void btn_save_Click(object sender, EventArgs e)
    {
        string imagename =
Path.GetFileName(file_image.PostedFile.FileName.ToString());
        file_image.SaveAs(Server.MapPath("../Assets/Cake/" +
imagename));

        string ins = "insert into
tbl_cake(cake_name,cake_description,cake_weight,caketype_id,cake_pri
ce,cake_image,baker_id) values('" + txt_name.Text + "','" +
txt_description.Text + "','" + txt_quantity.Text + "','" +
ddl_type.SelectedValue + "','" + txt_price.Text + "','" + imagename
+ "','" + Session["bid"] + "')";
        obj.ExecuteCommands(ins);
        Response.Write("<script>alert('New cake added')</script>");
        fillgrid();
        cancel();
    }
    public void filldrop()
    {
        obj.FillDrop(ddl_type, "caketype_name", "caketype_id",
"tbl_caketype");
    }
    public void fillgrid()
```

```
    {
        string sel = "select * from tbl_cake c inner join
tbl_caketype ct on c.caketype_id=ct.caketype_id  where c.baker_id='"
+ Session["bid"] + "'";
        obj.FillGridView(sel, grd_cake);
    }
    protected void grd_cake_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "gallery")
        {
            Session["cakeid"] = id;
            Response.Redirect("CakeGallery.aspx");
        }
        if (e.CommandName == "del")
        {
            string del = "delete from tbl_cake where cake_id='" + id
+ "'";
            obj.ExecuteCommands(del);
            string delete = "delete from tbl_cakegallery where
cake_id='" + id + "'";
            obj.ExecuteCommands(delete);
            fillgrid();
        }
        fillgrid();
    }
    public void cancel()
    {
        txt_name.Text = "";
        txt_description.Text = "";
        txt_quantity.Text = "";
        txt_price.Text = "";
```

```
        ddl_type.SelectedIndex=0;
    }
}
```

15.1.5 CCakeOrderStatus.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Customer_Default : System.Web.UI.Page
{
    bakers obj = new bakers();
    public static int id;
    public static int status;
    protected void Page_Load(object sender, EventArgs e)
    {
        fillgrd();
    }
    public void fillgrd()
    {
        string sel = "select * from tbl_customizecake cc inner join
tbl_flavour f on f.flavour_id=cc.flavour_id where ccake_status=0 and
cc.user_id='"+Session["cid"]+"'";
        obj.FillGridView(sel, GridView1);
        string sel1 = "select * from tbl_customizecake cc inner join
tbl_baker  b on cc.baker_id=b.baker_id inner join tbl_flavour f on
f.flavour_id=cc.flavour_id  where ccake_status=1 and cc.user_id='" +
Session["cid"] + "'";
        obj.FillGridView(sel1, GridView2);
```

```
    }
    protected void grd_accept_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "payment")
        {
            Session["ccid"] = id;
            Response.Redirect("CCakePayment.aspx");
        }
    }
    protected void GridView2_RowDataBound(object sender,
GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            HiddenField hid =
(HiddenField)e.Row.FindControl("HiddenField1");
            LinkButton payLink =
(LinkButton)e.Row.FindControl("lnk_payment");
            Label lblMsg = (Label)e.Row.FindControl("lblMessage");
            string sel = "select * from tbl_customizecake Where
ccake_id='" + hid.Value + "'";
            DataTable dt = obj.GetDataTable(sel);
            if (dt.Rows.Count > 0)
            {
                status =
Convert.ToInt32(dt.Rows[0]["ccake_pstatus"]);
            }

            if (status == 0)
            {
```

```
                payLink.Visible = true;
            }
            if (status == 1)
            {
                lblMsg.Text = "Payement Completed";
                payLink.Visible = false;
            }
        }
    }
    protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "del")
        {
            string del = "delete * from tbl_customizecake where
ccake_id='" + id + "'";
            obj.ExecuteCommands(del);
        }
    }
}
```

14.1.6 AcceptedBakerRequest.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Deliveryboy_Default : System.Web.UI.Page
{
```

```
    bakers obj = new bakers();
    public static int id;
    public static int status;
    protected void Page_Load(object sender, EventArgs e)
    {
        fillgrid();
    }
    public void fillgrid()
    {
        string sel = "select * from tbl_baker b inner join tbl_cake
c on b.baker_id=c.baker_id inner join tbl_cakebook cb on
c.cake_id=cb.cake_id inner join tbl_customer u on
u.user_id=cb.user_id inner join tbl_assignborder ao on
cb.cakebook_id=ao.cakebook_id where ao.deliveryboy_id='" +
Session["dbid"] + "' and ao.assignborder_status!=3";
        obj.FillGridView(sel, grd_cake);
        string sel1 = "select * from tbl_baker b inner join
tbl_product p on b.baker_id=p.baker_id inner join tbl_productbook pb
on p.product_id=pb.product_id inner join tbl_customer u on
u.user_id=pb.user_id inner join tbl_assignborder ao on
pb.productbook_id=ao.productbook_id where ao.deliveryboy_id='" +
Session["dbid"] + "'  and ao.assignborder_status!=3";
        obj.FillGridView(sel1, grd_product);
    }
    protected void grd_cake_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "picked")
        {
            string update = "update tbl_assignborder set
assignborder_status=1 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update);
```

```
            string update1 = "update tbl_cakebook set
cakebook_dstatus=2 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
        if (e.CommandName == "out")
        {
            string update = "update tbl_assignborder set
assignborder_status=2 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update);
            string update1 = "update tbl_cakebook set
cakebook_dstatus=3 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
        if (e.CommandName == "delivery")
        {
            string update = "update tbl_assignborder set
assignborder_status=3 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update);
            string update1 = "update tbl_cakebook set
cakebook_dstatus=4 where cakebook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
    }
```

```
    protected void grd_product_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        id = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "picked")
        {
            string update = "update tbl_assignborder set
assignborder_status=1 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update);
            string update1 = "update tbl_productbook set
productbook_dstatus=2 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
        if (e.CommandName == "out")
        {
            string update = "update tbl_assignborder set
assignborder_status=2 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update);
            string update1 = "update tbl_productbook set
productbook_dstatus=3 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
        if (e.CommandName == "delivery")
        {
            string update = "update tbl_assignborder set
assignborder_status=3 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update);
```

```
            string update1 = "update tbl_productbook set
productbook_dstatus=4 where productbook_id='" + id + "'";
            obj.ExecuteCommands(update1);
            Response.Write("<script>alert('Updated
successfully')</script>");
            fillgrid();
        }
    }
    protected void grd_product_RowDataBound(object sender,
GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            HiddenField hid =
(HiddenField)e.Row.FindControl("HiddenField1");
            LinkButton Picked =
(LinkButton)e.Row.FindControl("LinkButton5");
            LinkButton outd =
(LinkButton)e.Row.FindControl("LinkButton3");
            LinkButton delivery =
(LinkButton)e.Row.FindControl("LinkButton2");
            Label lblMsg = (Label)e.Row.FindControl("lblMessage");
            string sel = "select * from tbl_productbook Where
productbook_id='" + hid.Value + "'";
            DataTable dt = obj.GetDataTable(sel);
            if (dt.Rows.Count > 0)
            {
                status =
Convert.ToInt32(dt.Rows[0]["productbook_dstatus"]);
            }

            if (status == 1)
            {
```

```
                lblMsg.Text = "Pending";
                Picked.Visible = true;
                outd.Visible = false;
                delivery.Visible = false;
            }
            if (status == 2)
            {
                lblMsg.Text = "Picked";
                Picked.Visible = false;
                outd.Visible = true;
                delivery.Visible = false;
            }
            if (status == 3)
            {
                lblMsg.Text = "Out for delivery";
                Picked.Visible = false;
                outd.Visible = false;
                delivery.Visible = true;
            }
        }
    }
    protected void grd_cake_RowDataBound(object sender,
GridViewRowEventArgs e)
    {
        if (e.Row.RowType == DataControlRowType.DataRow)
        {
            HiddenField hid =
(HiddenField)e.Row.FindControl("HiddenField1");
            LinkButton Picked =
(LinkButton)e.Row.FindControl("LinkButton4");
            LinkButton outd =
(LinkButton)e.Row.FindControl("LinkButton3");
```

```
            LinkButton delivery =
(LinkButton)e.Row.FindControl("LinkButton1");
            Label lblMsg = (Label)e.Row.FindControl("lblMessage");
            string sel = "select * from tbl_cakebook Where
cakebook_id='" + hid.Value + "'";
            DataTable dt = obj.GetDataTable(sel);
            if (dt.Rows.Count > 0)
            {
                status =
Convert.ToInt32(dt.Rows[0]["cakebook_dstatus"]);
            }

            if (status == 1)
            {
                lblMsg.Text = "Pending";
                Picked.Visible = true;
                outd.Visible = false;
                delivery.Visible = false;
            }
            if (status == 2)
            {
                lblMsg.Text = "Picked";
                Picked.Visible = false;
                outd.Visible = true;
                delivery.Visible = false;
            }
            if (status == 3)
            {
                lblMsg.Text = "Out for delivery";
                Picked.Visible = false;
                outd.Visible = false;
                delivery.Visible = true;
            }
```

```
        }
    }
}
```

14.1.7 Complaint.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Customer_Default : System.Web.UI.Page
{
    bakers obj = new bakers();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fillddl();
        }
    }
    protected void btn_send_Click(object sender, EventArgs e)
    {
        string ins = "insert into tbl_complaint
(complainttype_id,complaint_title,complaint_data,complaint_date,comp
laint_status,user_id) values('" + ddl_type.SelectedValue + "','" +
txt_title.Text + "','" + txt_cmplt.Text + "','" +
DateTime.Now.ToShortDateString() + "',0,'" + Session["cid"] + "')";
        obj.ExecuteCommands(ins);
```

```
        Response.Write("<script>alert('Complaint added
successfully')</script>");
        cancel();
        Response.Redirect("ViewComplaintStatus.aspx");
    }
    public void fillddl()
    {
        obj.FillDrop(ddl_type, "complainttype_name",
"complainttype_id", "tbl_complainttype");
    }
    public void cancel()
    {
        txt_title.Text = "";
        txt_cmplt.Text = "";
        ddl_type.ClearSelection();
    }
}
```

14.1.8 Logout.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class Guest_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        if (!Page.IsPostBack)
        {
            Session.Abandon();

            string nextpage = "../Guest/Login.aspx";
            Response.Write("<script language='javascript'>");
            Response.Write("{");
            Response.Write("var backlen=history.length;");
```

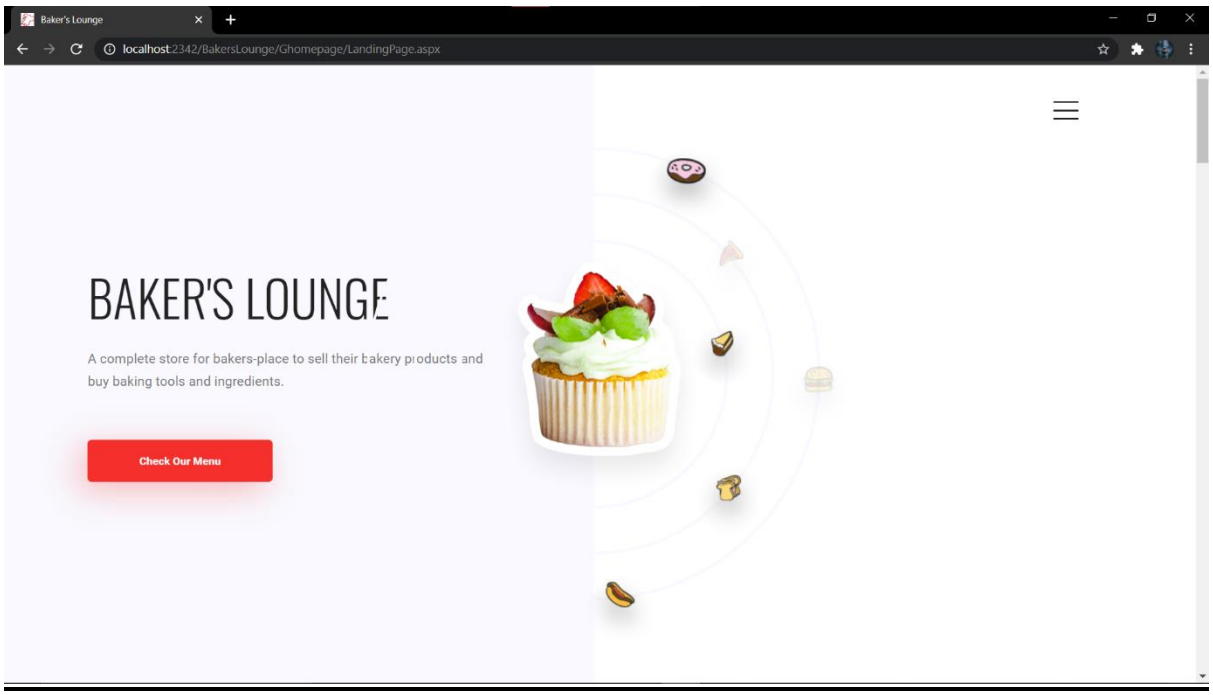
```
            Response.Write("history.go(-backlen);");
            Response.Write("window.location.href='" + nextpage +
"';");
            Response.Write("}");
            Response.Write("</script>");

        }
    }
 }
```

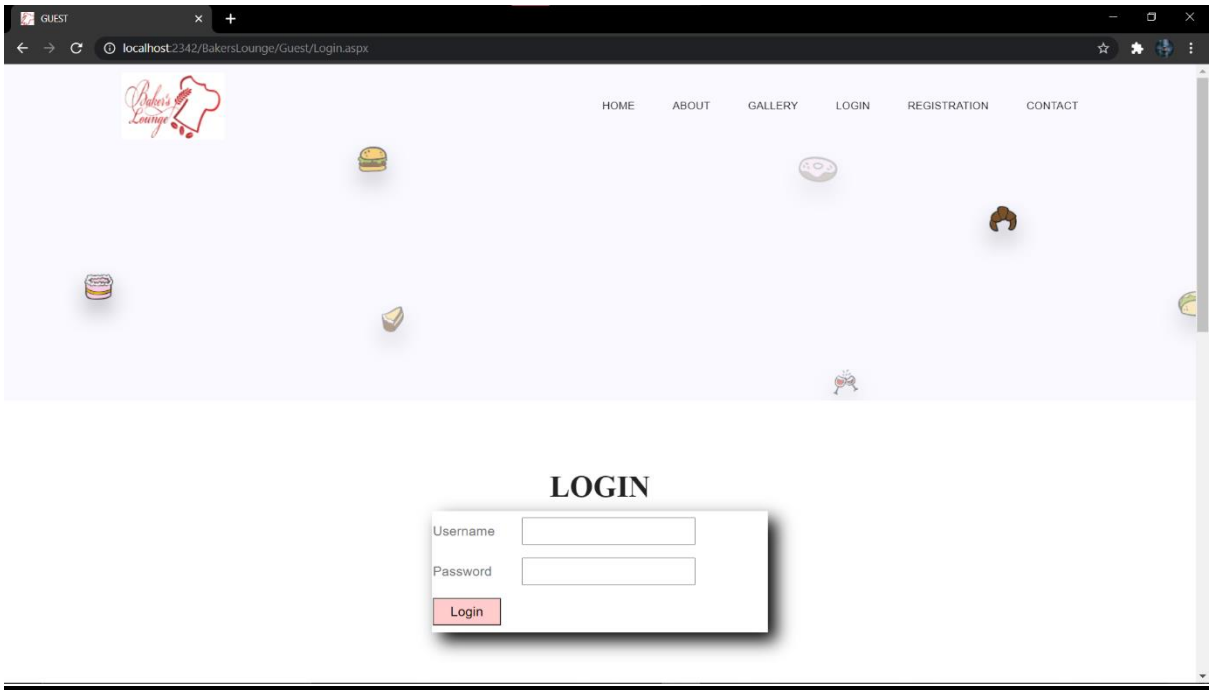
14.2 SCREENSHOTS

14.2.1 GUEST

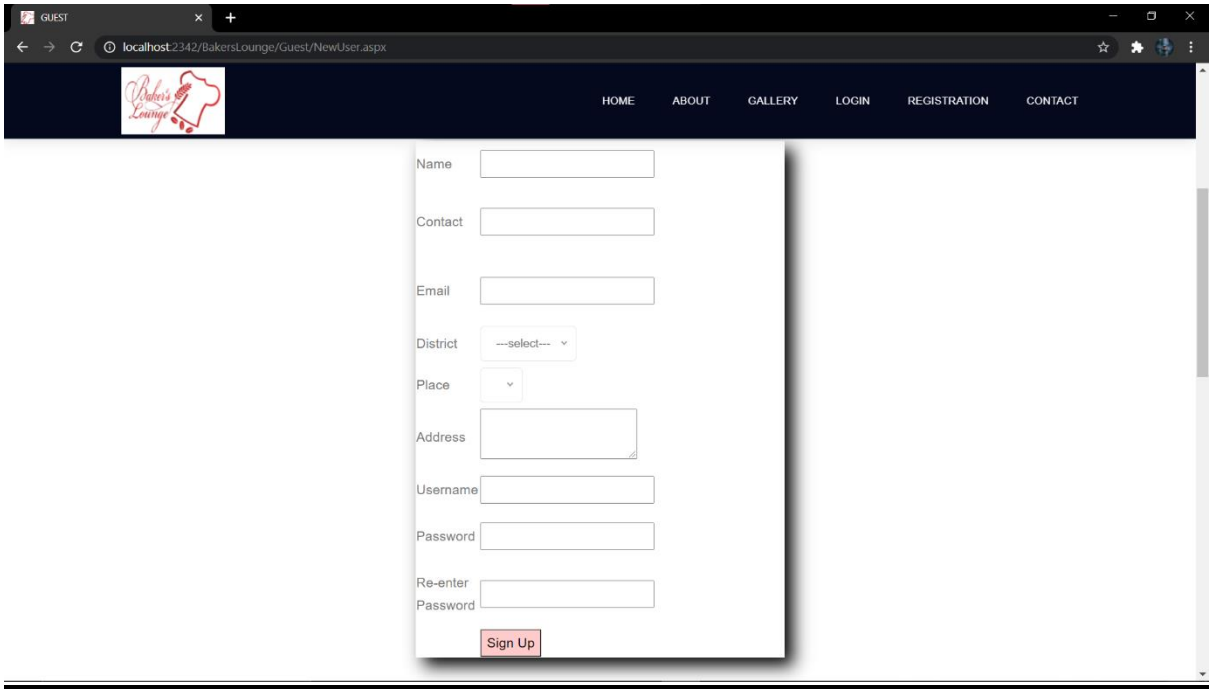
LandingPage



Login

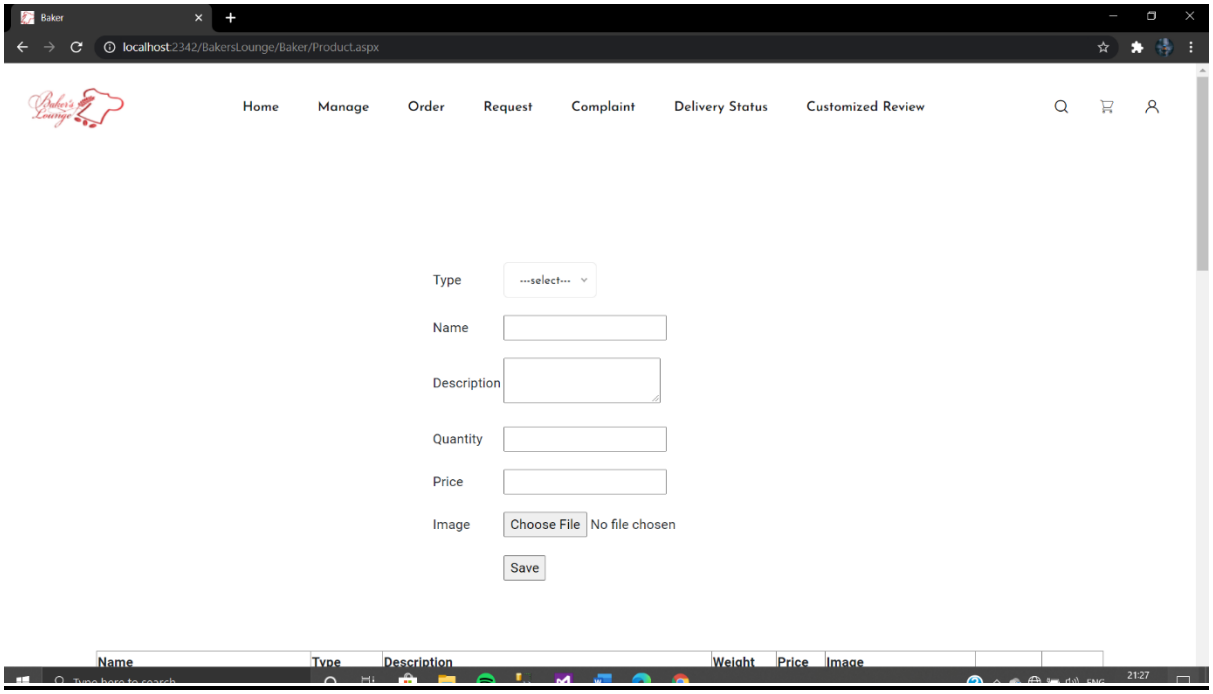


Users Registration

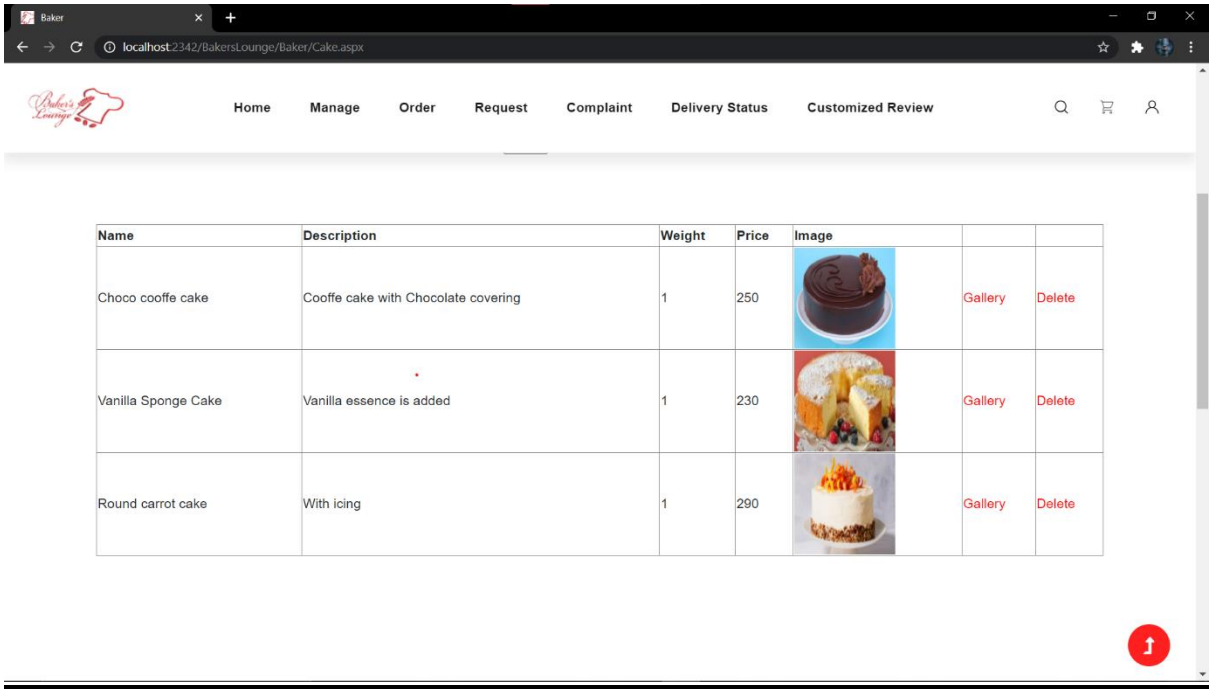


14.2.2 BAKER

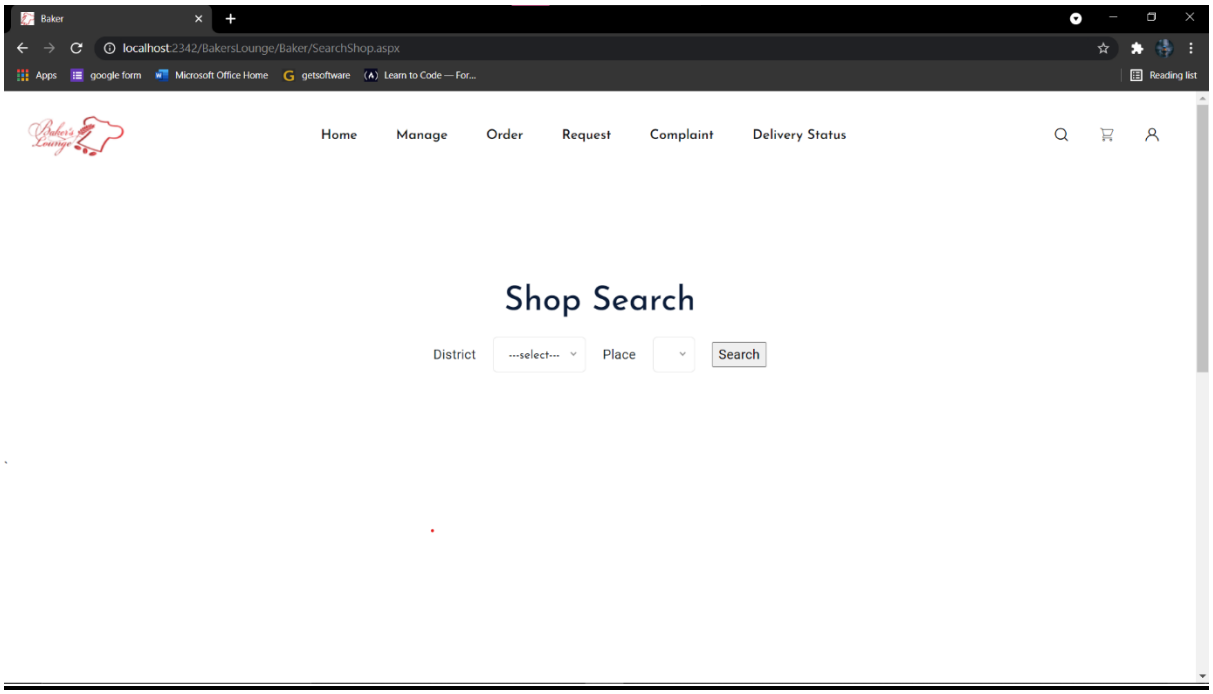
Product adding page



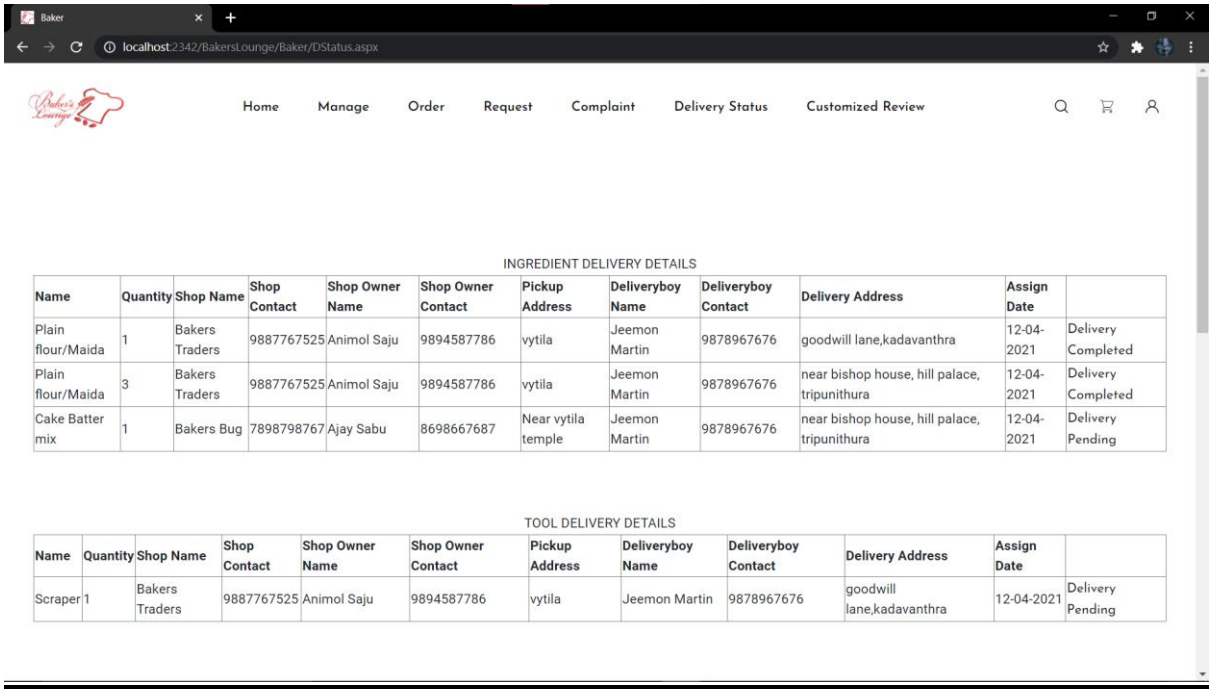
Cake managing page



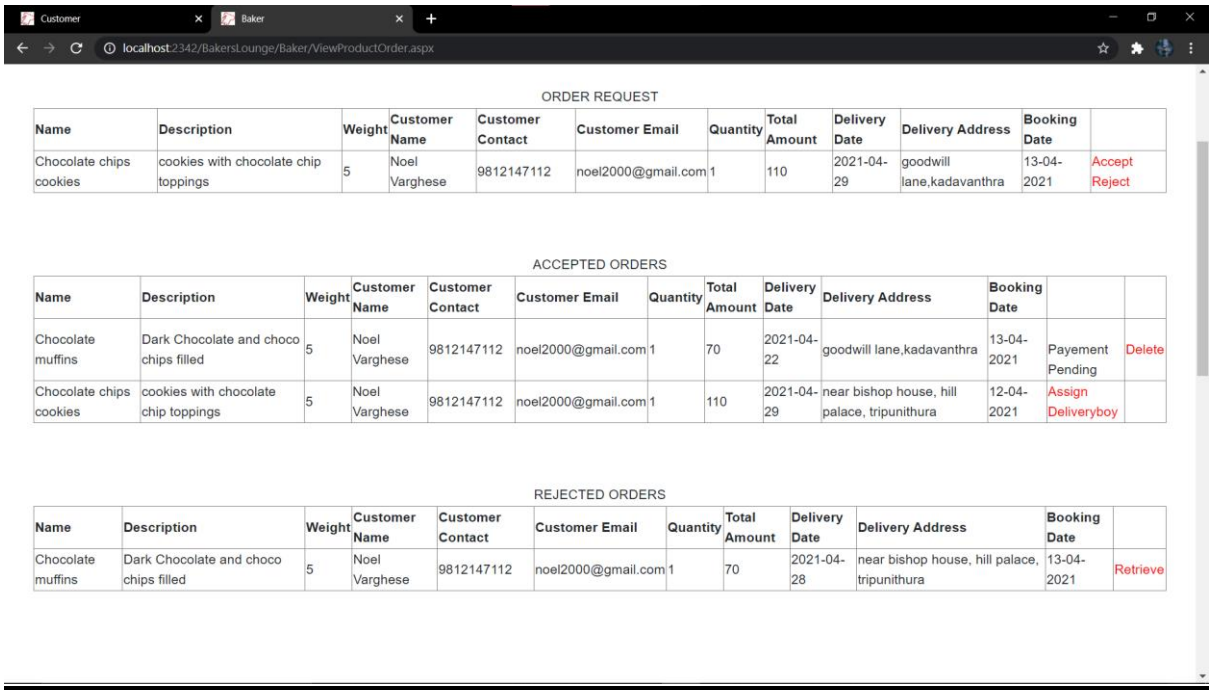
Search shop



Ingredient and tool delivery status page



Product booking details



ORDER REQUEST

Name	Description	Weight	Customer Name	Customer Contact	Customer Email	Quantity	Total Amount	Delivery Date	Delivery Address	Booking Date	
Chocolate chips cookies	cookies with chocolate chip toppings	5	Noel Varghese	9812147112	noel2000@gmail.com	1	110	2021-04-29	goodwill lane,kadavanthra	13-04-2021	Accept Reject

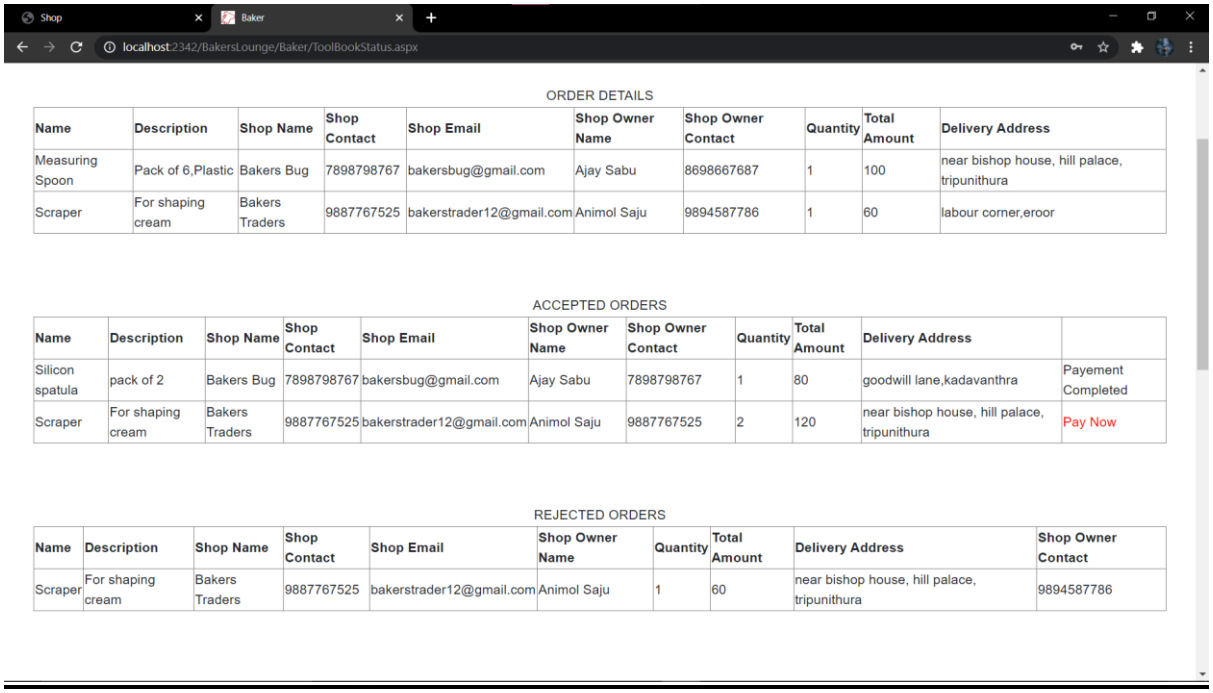
ACCEPTED ORDERS

Name	Description	Weight	Customer Name	Customer Contact	Customer Email	Quantity	Total Amount	Delivery Date	Delivery Address	Booking Date		
Chocolate muffins	Dark Chocolate and choco chips filled	5	Noel Varghese	9812147112	noel2000@gmail.com	1	70	2021-04-22	goodwill lane,kadavanthra	13-04-2021	Payement Pending	Delete
Chocolate chips cookies	cookies with chocolate chip toppings	5	Noel Varghese	9812147112	noel2000@gmail.com	1	110	2021-04-29	near bishop house, hill palace, tripunithura	12-04-2021	Assign Deliveryboy	

REJECTED ORDERS

Name	Description	Weight	Customer Name	Customer Contact	Customer Email	Quantity	Total Amount	Delivery Date	Delivery Address	Booking Date	
Chocolate muffins	Dark Chocolate and choco chips filled	5	Noel Varghese	9812147112	noel2000@gmail.com	1	70	2021-04-28	near bishop house, hill palace, tripunithura	13-04-2021	Retrieve

Tool booking status page



ORDER DETAILS

Name	Description	Shop Name	Shop Contact	Shop Email	Shop Owner Name	Shop Owner Contact	Quantity	Total Amount	Delivery Address
Measuring Spoon	Pack of 6,Plastic	Bakers Bug	7898798767	bakersbug@gmail.com	Ajay Sabu	8698667687	1	100	near bishop house, hill palace, tripunithura
Scraper	For shaping cream	Bakers Traders	9887767525	bakerstrader12@gmail.com	Animol Saju	9894587786	1	60	labour corner,eroor

ACCEPTED ORDERS

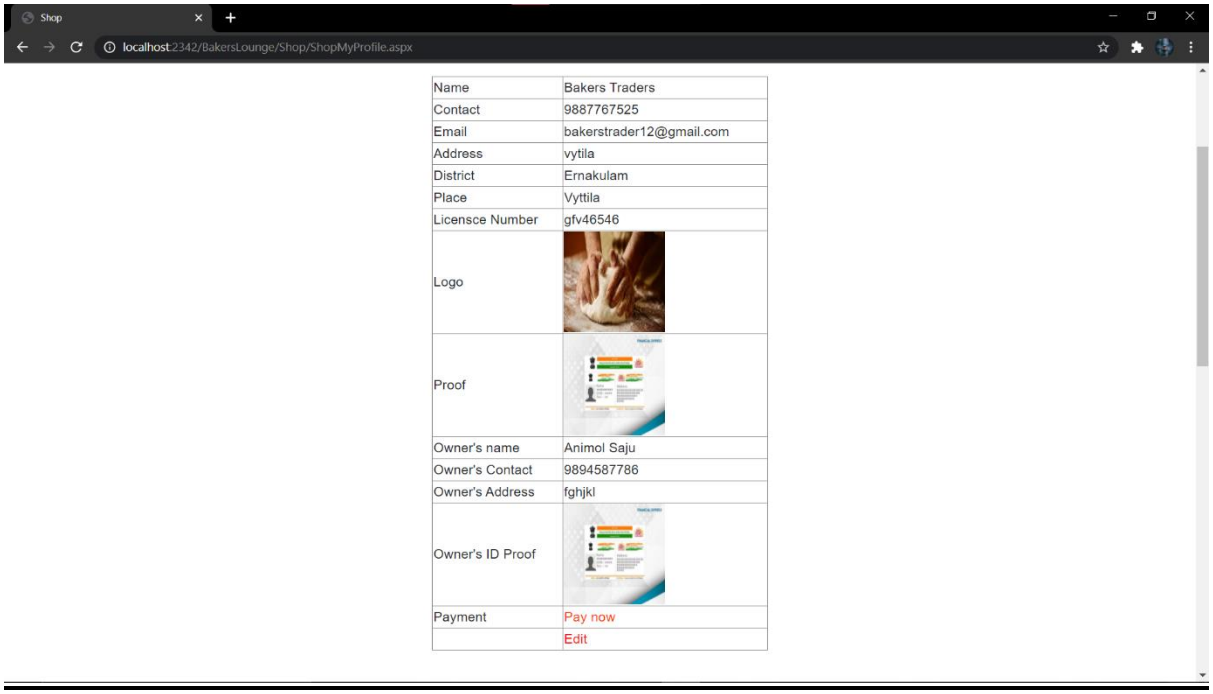
Name	Description	Shop Name	Shop Contact	Shop Email	Shop Owner Name	Shop Owner Contact	Quantity	Total Amount	Delivery Address	
Silicon spatula	pack of 2	Bakers Bug	7898798767	bakersbug@gmail.com	Ajay Sabu	7898798767	1	80	goodwill lane,kadavanthra	Payement Completed
Scraper	For shaping cream	Bakers Traders	9887767525	bakerstrader12@gmail.com	Animol Saju	9887767525	2	120	near bishop house, hill palace, tripunithura	Pay Now

REJECTED ORDERS

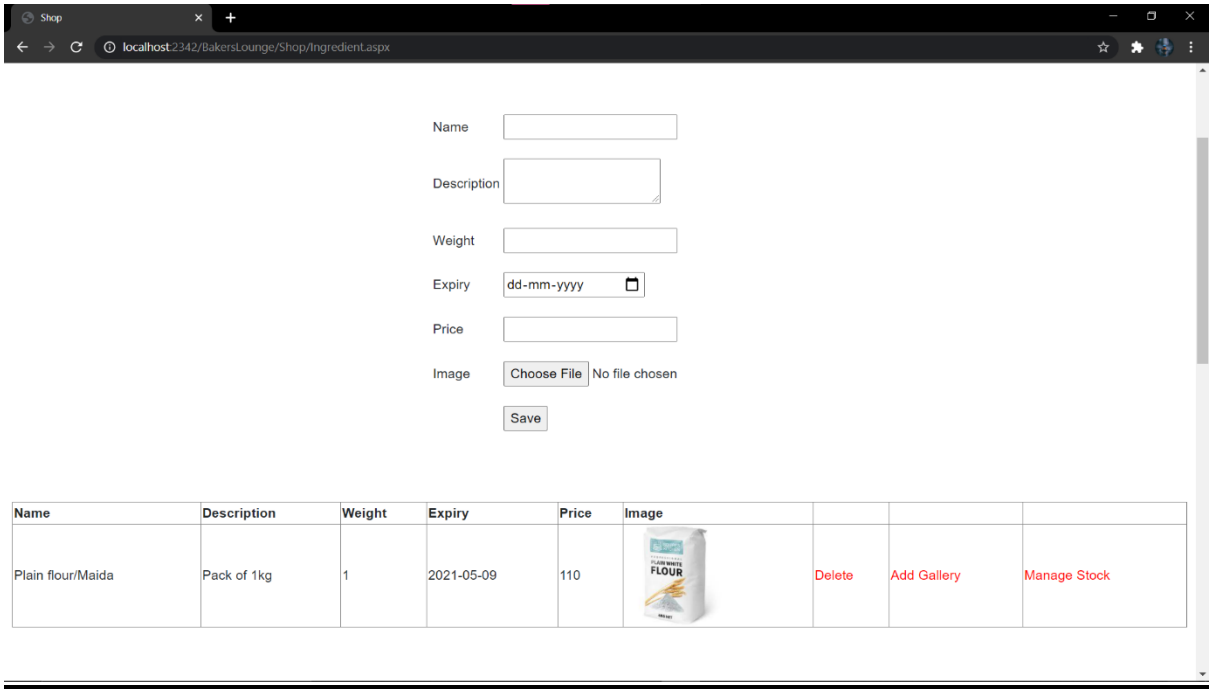
Name	Description	Shop Name	Shop Contact	Shop Email	Shop Owner Name	Quantity	Total Amount	Delivery Address	Shop Owner Contact
Scraper	For shaping cream	Bakers Traders	9887767525	bakerstrader12@gmail.com	Animol Saju	1	60	near bishop house, hill palace, tripunithura	9894587786

14.2.3 SHOP

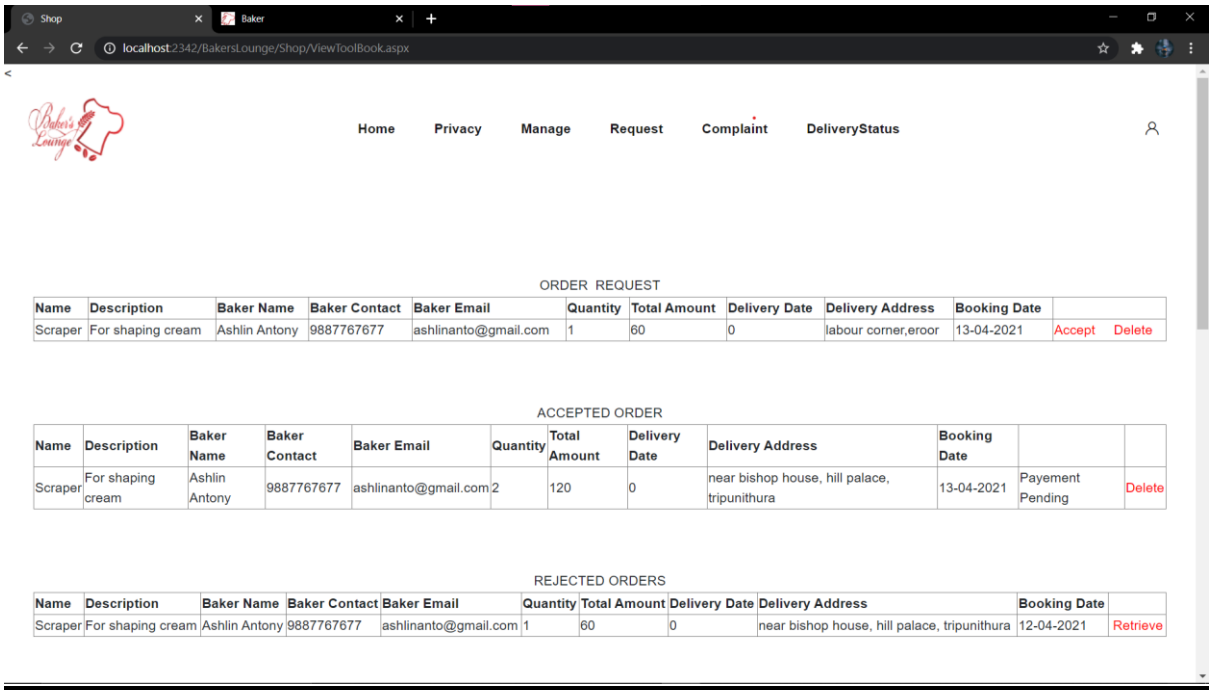
View profile page



Ingredient page

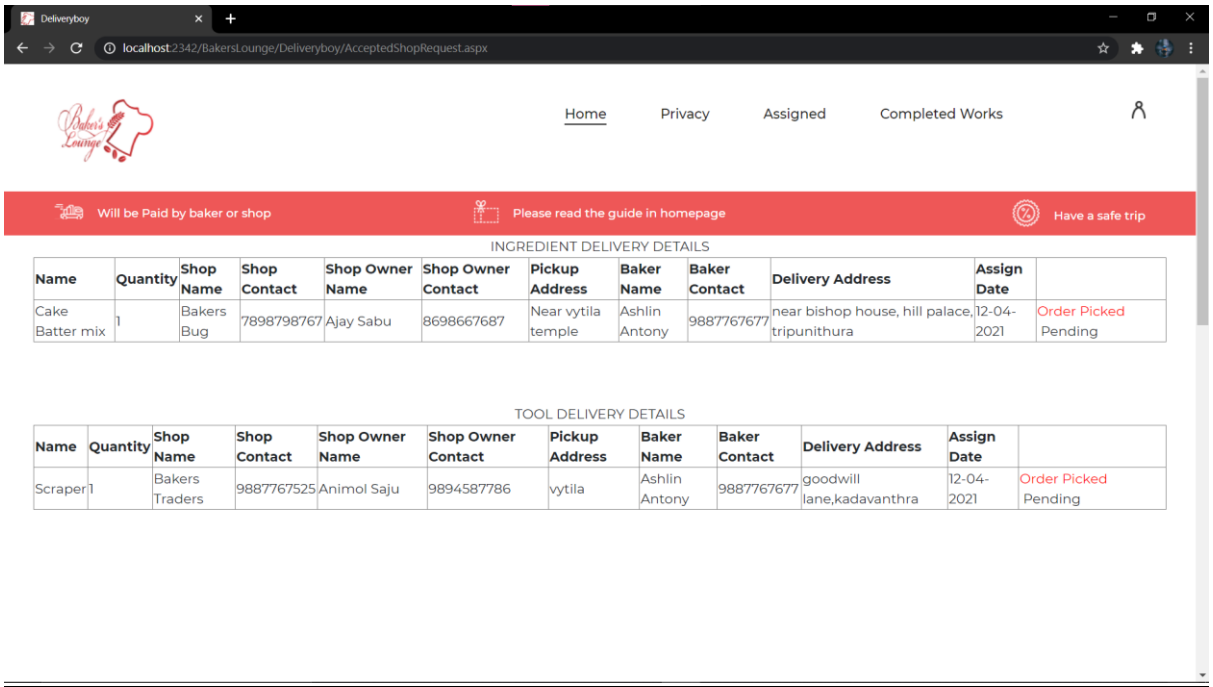


Tool book details page

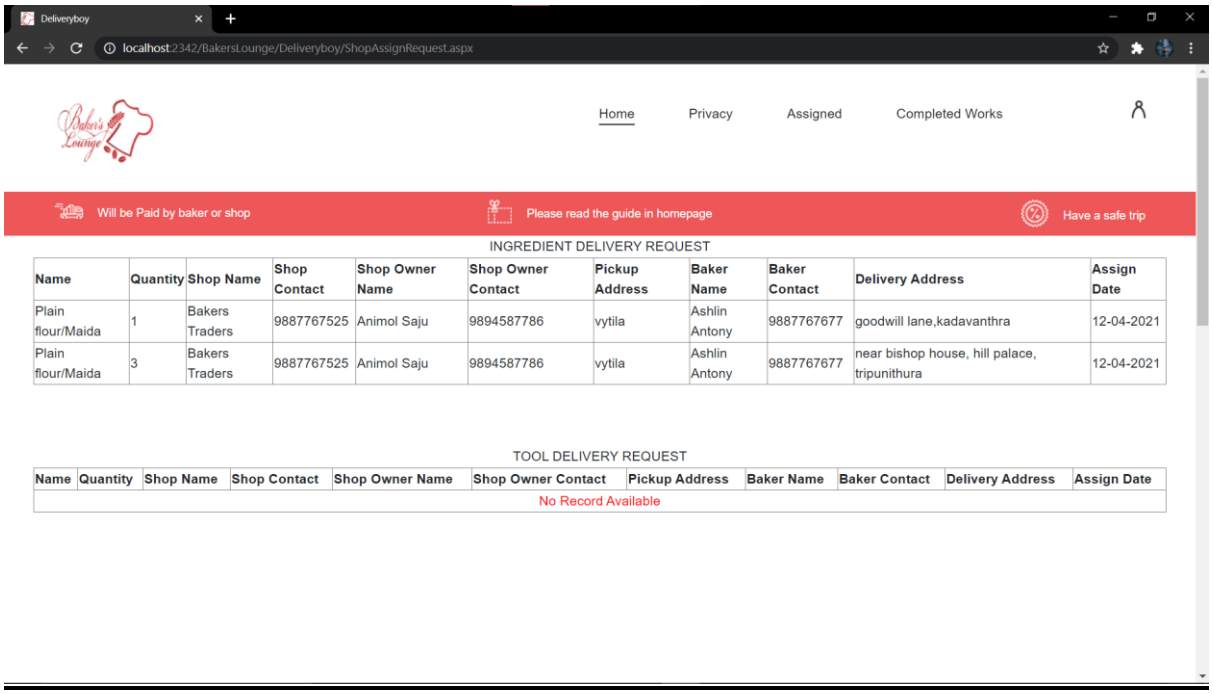


14.2.4 DELIVERYBOY

Status update page for shop assigned works

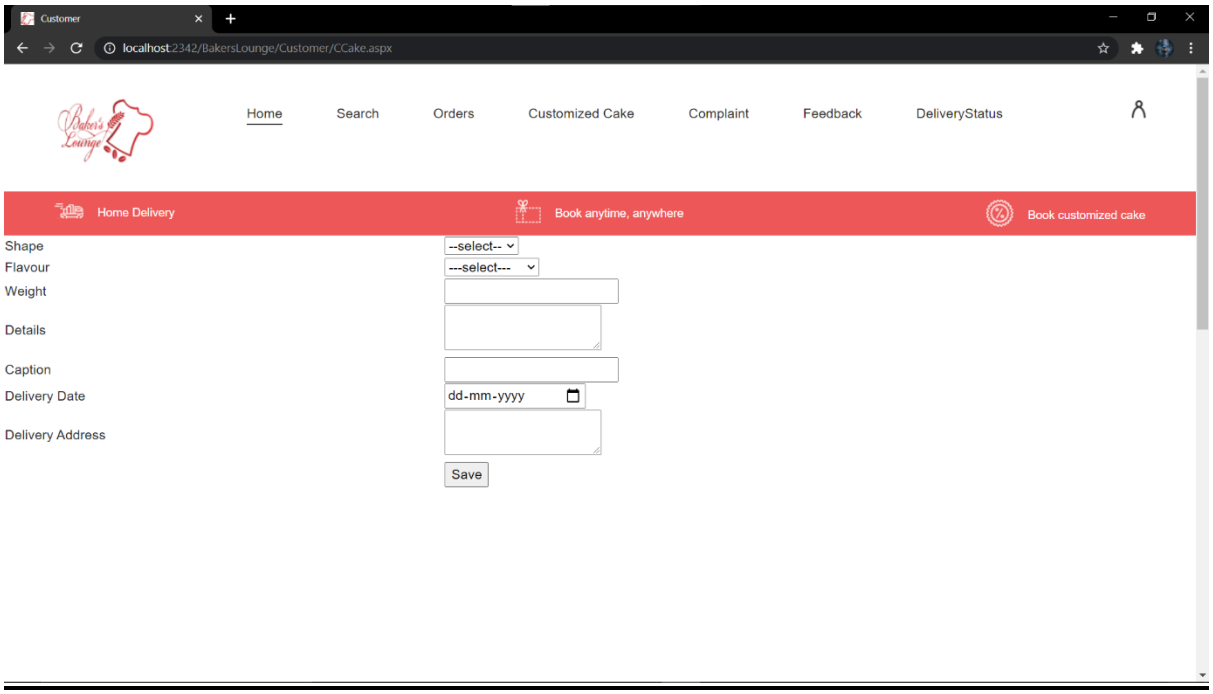


Completed shop assigned work page

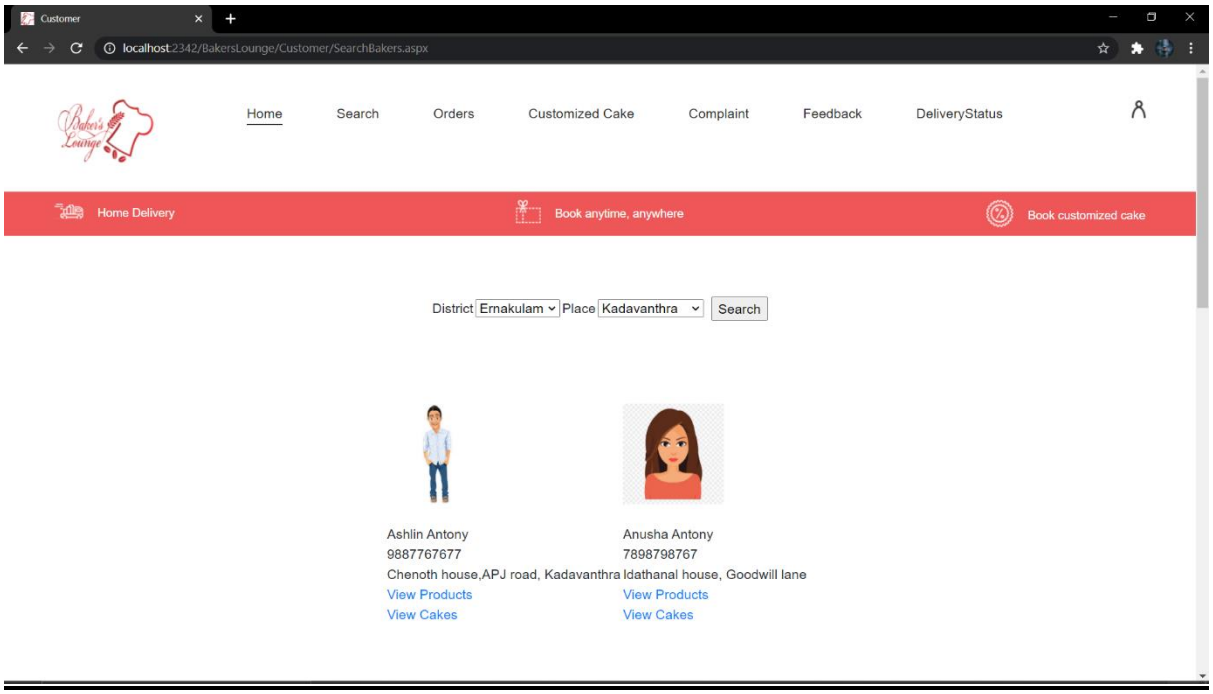


14.2.5 CUSTOMER

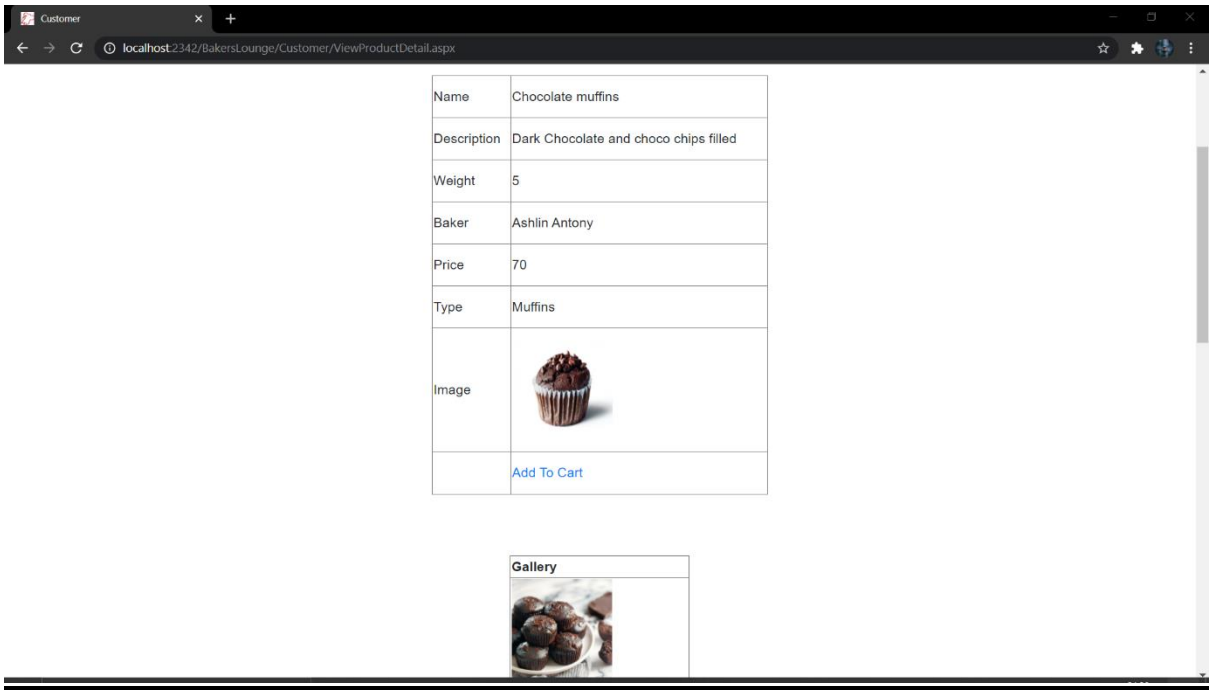
Customized cake booking page



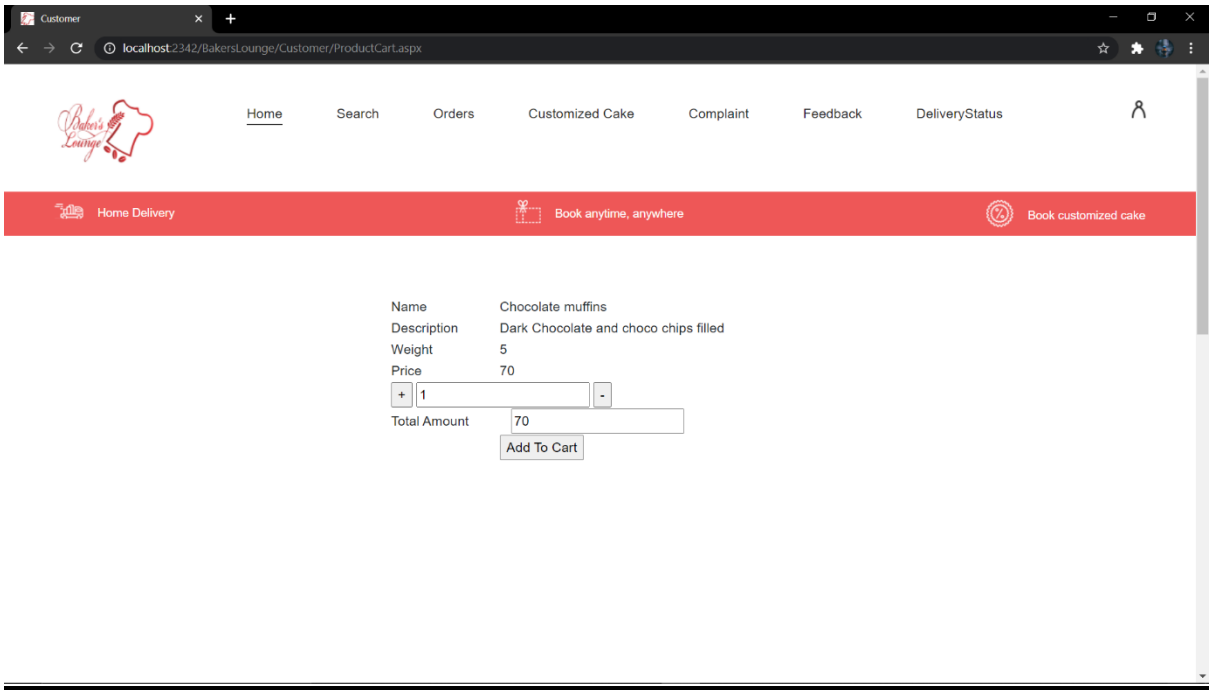
Search bakers page



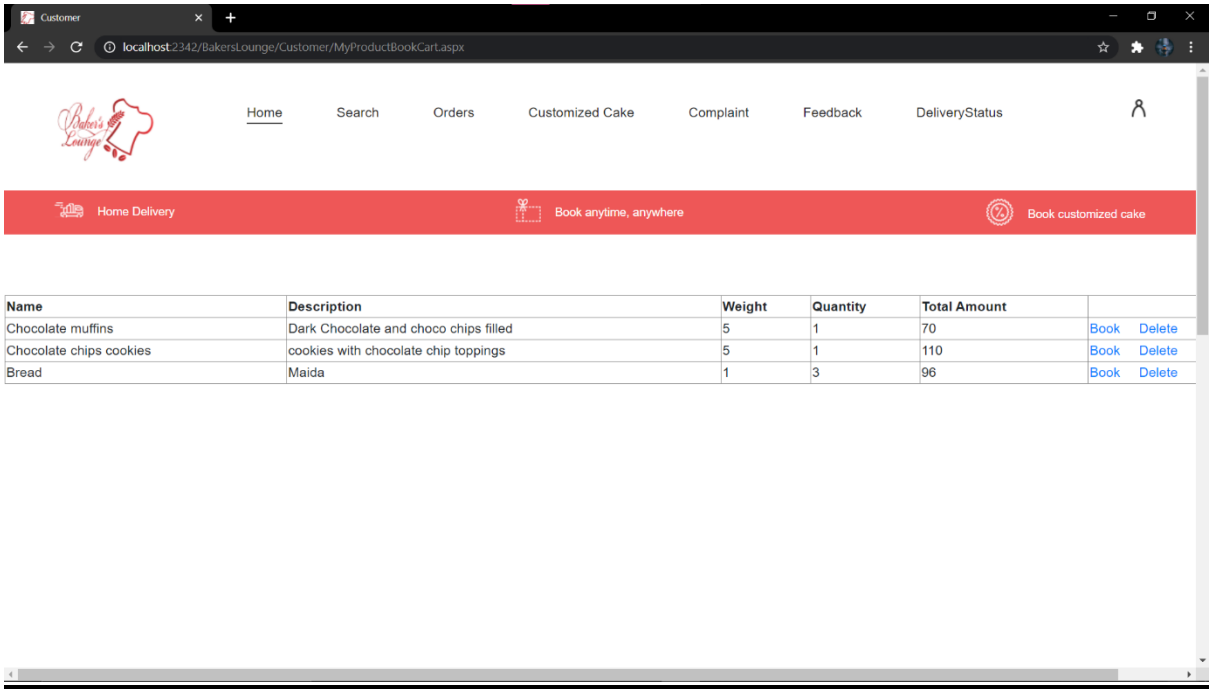
Search product detail page



Product add to cart page



Product cart page



Name	Description	Weight	Quantity	Total Amount	
Chocolate muffins	Dark Chocolate and choco chips filled	5	1	70	Book Delete
Chocolate chips cookies	cookies with chocolate chip toppings	5	1	110	Book Delete
Bread	Maida	1	3	96	Book Delete

Product booking page