**CHAPTER 1**

**INTRODUCTION**

**1.1 Overview**

Digital menu allows diners to explore new items and try them. It gives them the confidence on what to have. Special items of the day or offers can be instantly presented to grab more order value. Since the staff is focused on delivery of order, it increases the table turnaround ratio too that adds to the sales figures.

The number of meals eaten outside the home continues to increase; the restaurant and cafe industries are one of the fast growing service industries in India. With growing competitiveness in them and the similarity of products or services offered by them, it has become increasingly important to identify the factors that determine the basis upon which customers choose between. The issue of how customers select restaurants and cafes has been given considerable attention by both researchers and practitioners. However, surprisingly little research has been generated on the criteria customers use in order to select. Furthermore, previous studies have not considered the extent to which important choice criteria would differ depending on the restaurant and cafe type. It is also important to note that customers with different characteristics tend to use different criteria in choosing restaurants and cafes.

The traditional system is a restaurant paper menu and ordering system is replaced with an electronic medium i.e. digital menu. Due to a digitalized system, the risk of manual errors is eliminated, thus eliminating the communication barrier. The digital menu displays all the information the customer needs to know the order he has placed. This self-service fast food restaurant will be equipped with a user-friendly digital menu. For this system there will be a system administrator who will have the rights to enter the menu with their current prevailing prices. He/she can enter anytime in the system by a secured system password to change the menu contents by adding or deleting an item or changing its price.

Now when the customer enters the restaurant, he will place his order of the digital menu using intuitive graphical user interface, right from the selection of language till payment confirmation. He will select from the food options according to his choice and the system will display the payment amount he has to make once he has finished with his order.

**1.2 Project Scope and Objective**

* Item modification option
* Price modification option
* Added inventory staff
* Detail of customer
* Daily and yearly report

**Objective**

This main objective of this system is to kill the way time of the customer by enhancing the restaurant’s automated dining experience and to improve the services of the restaurant by lessening the work load of waiter staff.

Main aim of digital menu system is to develop an online application for the users that will help them to order food in the restaurants. The chef of the restaurant can also order the required ingredients by using our application.

**1.3 Existing System**

In the existing system, the restaurants have paper based system through which food order is taken. The chef has to inform the restaurant manager about the ingredients availability manually. This system is time consuming and includes lot of manual stress. This system is not effective and accurate.

**Limitations of the Existing System**

* Time consuming
* Manual Process
* Tedious task
* Ambiguous approach

**1.4 Proposed System**

We propose a system to develop an application which will enable the users to select the menu in the restaurant by booking option. They can choose menu of their interest. Based on the booking id of the particular customer, the food is served. The chef of the restaurant can order the ingredients by using this application. The farmer will supply the ingredients to the retailer, then in turn the retailers will deliver the ingredients to the restaurants based on booking.

**Advantages:**

* This system is accurate and time saving
* This system is cost-effective
* Reduces waiting time

**1.5 Benefits of Digital Table Menu**

* Easily changes or updates the important information on the menu
* Adding new items on the menu and changing the price of items is now easy
* Reduce perceived way times and enhances customer dining experience
* Eliminates the traditional systems of taking orders and submitting bills
* It can display emergency messages quickly
* It eliminates recurring cost associated with printing of paper menus

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Introduction**

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

When we write a literature review in respect of your project, you have to write the researches made by various analysts - their methodology (which is basically their abstract) and the conclusions they have arrived at.

The purpose of a literature review is to gain an understanding of the existing research and debates relevant to a particular topic or area of study, and to present that knowledge in the form of a written report. Conducting a literature review helps us build knowledge in our field. We will learn about important concepts, research methods, and experimental techniques that are used in our area of research. We also gain insight into how researchers apply the concepts to real world problems. Another great benefit of literature reviews is that as we read, we will get a better understanding of how research findings are presented and discussed in our particular discipline.

**2.2 Paper Survey**

“Begin ASP.NET 2.0 with visual C#.NET”, Wrox, By Chris Ullman”

**Title:** Stenography using Image Audio Video

**Date of Publish:** 2018

**Author:** Sunil Kumar N. & K. M. Sowmyashree

**Introduction**

Image is an electronic medium for copying, playback, broadcasting and display of moving visual media. Image security has gained importance over time in numerous applications wherein information in the form of image is to be secured from unauthorized user. Image consisting of several frames which is nothing but images. The use of internet has increased tremendously over the years and the concept of data security is gaining momentum. The word steganography combines the Greek words steganos meaning “covered” and graphic meaning “writing”. The art and science of hiding information by embedding messages within other is steganography. It works by replacing bits of useless or unused data. Steganography is an Encryption technique that can be used along with cryptography as an extra-secure method in which to protect data. It can be applied to images, a image file or an audio file. Steganography is used to supplement encryption. An encrypted file may still hide information, by using steganography even if the encrypted file is deciphered, the hidden message is not seen.

Every Software development requires the survey process. The Survey process is needed to get the requirement for the software. The Survey also consists of studying the present system and also studying about the tools needed for the development of the software. A proper understanding of the tools is very much essential. Following an extract of the information of the immaterial collected during literature survey. In this paper, an introductory look at information hiding techniques and historical details is discussed. Several methods for hiding data in audio, image is described with appropriate to environment of each medium as well as strength and weakness of each medium.

The information about secret key, transmission protocol, computer file system, hiding techniques are discussed. In this paper, the different types of steganography methods its pros and cons are discussed in detail. It gives information about efficient method for sending safely to this destination. The use of steganography application is to hide different types of data within cover file. This is done according to the embedding algorithm and a secret key that performs the actions of embedding process. In this paper, the image data embedding scheme is proposed. We can replace one or more LSB of each pixel in image frame. It becomes very difficult for the intruder to guess the data hidden in a frame. An advanced data hiding method by using different bit with help of LSB substitution is proposed and analysed. In this paper, it explains the prime need of hiding data from eavesdroppers is accomplished by the use of steganography.

It explains about the wide researches that have been carried out on image steganography due to high capacity of information been stored in image file.

This paper presented using LSB insertion which is very efficient method to embed data into a cover medium. It has explained the LSB insertion method for image steganography and its application. In this paper the focus on the data security approach with combined encryption and steganography techniques for secret communication by hiding it inside a multimedia files is done. The file such as images, audio, image contains collection of bits that can be further translated into same.

The files composed of insignificant bits or unused areas which can be used for overwriting of other data. This paper explains the proposed algorithm using image steganography for enhancing data security. In this paper, the explanation on combination of cryptography and steganography is used for data hiding in image clips. A random frame selection, pixel swapping, and encryption of message has been done to enhance security of secret information which goes under the cover of image clips. Image steganography method has been developed to transfer secret data. In this paper, the modern secure image steganography presents a challenging task of transferring embedded information to destination without being detected. Here, a simple approach for embedding message into image or the image from pixel of carrier image is replaced with message information so that it cannot be observed by human visual system, therefore exploits some limitations of human visual system.

“ASP.net complete reference - Pearson Publication”

**Title:** Restaurant scheduler

**Date of Publish:** 2015

**Author:** Sachin M

**Introduction**

In the fall of 2004, at a popular software conference I realized how all major component vendors were advertising their ASP.NET products using a new word Ajax. Only a few weeks later, a brand new module in my popular ASP.NET master class made its debut using Ajax to improve the user experience. At its core, Ajax is a little thing and fairly old too as I presented the engine of it (XmlHttpRequest) to a C++ audience at TechEd 2000, only four weeks before the public announcement of the .NET platform.

As emphatic as it may sound, that crazy little thing called Ajax changed the way we approach Web development. Ajax triggered a chain reaction in the world of the Web. Ajax truly represents paradigm shift for Web applications. And, as the history of science proves, a paradigm shift always has a deep impact, especially in scenarios that were previously stable and consolidated. We are now really close to the day we will be able to say “the Web” without feeling the need to specify whether it contains Ajax or not. Just the Web which has a rich client component, a made-to-measure layer of HTTP endpoints to call, and interchangeable styles.

Like it or not, the more we take the Ajax route, the more we move away from ASP.NET Web Forms. In the end, it’s just like getting older. Until recently, Web Forms was a fantastic platform for Web development. The Web, however, is now going in a direction that Web Forms can’t serve in the same stellar manner.

It’s not yet time to cease ASP.NET Web Forms development. However, it’s already time for you to pay a lot more attention to aspects of Web development that Web Forms specifically and deliberately shielded you from for a decade CSS, JavaScript, and HTML mark-up.

In ASP.NET master class, I have a lab in which I first show how to display a data-bound grid of records with cells that trigger an Ajax call if clicked. I do that in exactly the way one would do it as an ASP.NET developer. Next, I challenge attendees to rewrite it without inline script and style settings. And yes a bit perversely I also tell anyone who knows jQuery not to use it. The result is usually a thoughtful and insightful experience, and the code I come up with gets better every time. ASP.NET Web Forms is not dead, no matter what ASP.NET MVC the twin technology can become. But it’s showing signs of age. As a developer, you need to recognize that and revive it through robust injections of patterns, JavaScript and jQuery code, and Ajax features.

We left out some of the classic topics you found in earlier versions, such as ADO.NET and even LINQ-to-SQL. I also reduced the number of pages devoted to controls. I brought in more coverage of ASP.NET underpinnings, ASP.NET configuration, jQuery, and patterns and design principles. Frankly, not a lot has changed in ASP.NET since version 2.0.

“Programming in C#, 3E” by BalagurusamyTata McGraw-Hill Education, 2010

**Title:** Programming in C#, 3E

**Date of Publish:** 1991

**Author:** Balagurusamy

**Introduction**

Microsoft has 'bet the company' on .NET. When a company of their size and influence spends billions of dollars and reorganizes its entire corporate structure to support a new platform, it is reasonable for programmers to take notice. It turns out that .NET represents major change in the way you'll think about programming. It is, in short, a new development platform designed to facilitate object-oriented Internet development. The programming language of choice for this object-oriented Internet-centric platform is C#, which builds on the lessons learned from C (high performance), C++ (object-oriented structure), Java (garbage collected, high security), and Visual Basic (rapid development) to create a new language ideally suited for developing component-based n-tier distributed web applications.

The goal of C# is to provide a simple, safe, modern, object-oriented, Internet-centric, high-performance language for .NET development. C# is a new language, but it draws on the lessons learned over the past three decades. In much the way that you can see in young children the features and personalities of their parents and grandparents, you can easily see in C# the influence of Java, C++, Visual Basic (VB), and other languages.

The focus of this book is the C# language and its use as a tool for programming on the .NET platform. In my primers on C++, We advocate learning the language first, without regard to Windows or Unix programming. With C# that approach would be pointless. You learn C# specifically to create .NET applications; pretending otherwise would miss the point of the language. Thus, this book does not consider C# in a vacuum but places the language firmly in the context of Microsoft's .NET platform and in the development of desktop and Internet applications.

**2.3 Technological Survey**

**2.3.1 Why .NET is used?**

* The Microsoft .Net Framework is a platform that provides tools and technologies you need to build Networked Applications as well as Distributed Web Services and Web Applications.
* The .Net Framework provides the necessary compile time and run-time foundation to build and run any language that conforms to the Common Language Specification (CLS).
* The main two components of .Net Framework are Common Language Runtime (CLR) and .Net Framework Class Library (FCL).
* The Common Language Runtime (CLR) is the runtime environment of the .Net Framework that executes and manages all running code like a Virtual Machine.
* The .Net Framework Class Library (FCL) is a huge collection of language-independent and type-safe reusable classes.
* The .Net Framework Class Libraries (FCL) are arranged into a logical grouping according to their functionality and usability is called Namespaces.
* Microsoft .Net Languages Source Code are compiled into Microsoft Intermediate Language (MSIL). MSIL we can call it as Intermediate Language (IL) or Common Intermediate Language (CIL). Microsoft Intermediate Language (MSIL) is a CPU independent set of instructions that can be converted to the native code. Metadata also created in the course of compile time with Microsoft Intermediate Language (MSIL) and stored it with the compiled code.
* Metadata is completely self-describing. Metadata is stored in a file called Manifest, and it contains information about the members, types, references and all the other data that the Common Language Runtime (CLR) needs for execution. The Common Language Runtime (CLR) uses metadata to locate and load classes, generate native code, provide security, and execute Managed Code. Both Microsoft Intermediate Language (MSIL) and Metadata assembled together are known as Portable Executable (PE) file. Portable Executable (PE) is supposed to be portable across all 32-bit operating systems by Microsoft .Net Framework.
* During the runtime the Common Language Runtime (CLR)'s Just In Time (JIT) compiler converts the Microsoft Intermediate Language (MSIL) code into native code to the Operating System. The native code is Operating System independent and this code is known as Managed Code, that is, the language's functionality is managed by the .NET Framework.
* The Common Language Runtime (CLR) provides various Just In Time (JIT) compilers, and each works on a different architecture depends on Operating Systems, that means the same Microsoft Intermediate Language (MSIL) can be executed on different Operating Systems.
* The .NET framework is a great choice when it comes to deliver multiple web based applications as per the latest market trends.
* It makes complex application looks easier and enables development of both web based and desktop based application in a much better and faster way.
* With the help of its versatile and dynamic library it ensures top level security and rapid development at the same time. .NET is language neutral. The three most common languages in .NET are C#, Visual Basic, and Delphi. But many others exist for .NET as well, including FORTRAN, Smalltalk, and others. .
* But .NET is more than just language neutral. .NET brings all the languages together through its CLR (Common Language Runtime) and CTS (Common Type System).
* The CLR and CTS allow all the languages to use assemblies produced by other languages as if they were produced by the same language.
* There is no more awkward translation of parameter types, calling conventions, or naming conventions. Now, C# users can use all code produced by Delphi programmers, Visual Basic users can use all code produced by C# developers, and any combination of languages. Some beneficial use of .NET framework:
* It significantly decreases the quantity of code necessary in large web applications which are developed in .NET framework.
* Web applications developed in [ASP.NET](http://asp.net/) are secure as Windows confirmation and configuration can be attained for every application.
* This development provides WYSIWYG (What You See Is What You Get).
* It provides server controls and blueprints with capability of drag and drop and involuntary operation.
* HTML code and source code are separated so changes can be done easily in[ASP.NET](http://asp.net/) framework.

**2.3.2 ASP.NET:**

ASP.NET is an [open-source](https://en.wikipedia.org/wiki/Open_source) [server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) designed for [web development](https://en.wikipedia.org/wiki/Web_development) to produce [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page). It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service).

ASP.NET is built on the [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](https://en.wikipedia.org/wiki/List_of_CLI_languages). The ASP.NET [SOAP](https://en.wikipedia.org/wiki/SOAP) extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is [ASP.NET Core](https://en.wikipedia.org/wiki/ASP.NET_Core). It is a re-implementation of ASP.NET as a modular [web framework](https://en.wikipedia.org/wiki/Web_framework), together with other frameworks like [Entity Framework](https://en.wikipedia.org/wiki/Entity_Framework). The new framework uses the new open-source [.NET Compiler Platform](https://en.wikipedia.org/wiki/.NET_Compiler_Platform) and is [cross platform](https://en.wikipedia.org/wiki/Cross_platform). [ASP.NET MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC), ASP.NET Web API, and ASP.NET Web Pages.

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while a web sites structure allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with a ".aspx" extension; these files typically contain static ([X](https://en.wikipedia.org/wiki/XHTML))[HTML](https://en.wikipedia.org/wiki/HTML) markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page.

**2.3.3 Introduction to C#**

With the introduction of the .NET framework, Microsoft included a new language called C# (pronounced C Sharp). C# is designed to be a simple, modern, general-purpose, object-oriented programming language, borrowing key concepts from several other languages, most notably Java.

C# could theoretically be compiled to machine code, but in real life, it's always used in combination with the .NET framework. Therefore, applications written in C#, requires the .NET framework to be installed on the computer running the application. While the .NET framework makes it possible to use a wide range of languages, C# is sometimes referred to as THE .NET language, perhaps because it was designed together with the framework.   
 C# is an Object Oriented language and does not offer global variables or functions. Everything is wrapped in classes, even simple types like int and string, which inherits from the System. Object class.

 C# is intended to be the premier language for writing NGWS (Next Generation Windows Services) applications in the enterprise computing space. The programming language C# derives from C and C++; however, it is modern, simple, entirely object-oriented, and type-safe. If you are a C/C++ programmer, you’re learning curve will be flat.

Contributing to the ease of use is the elimination of certain features of C++: no more macros, no templates, and no multiple inheritance. The aforementioned features create more problems than they provide benefit—especially for enterprise developers.

New features for added convenience are strict type safety, versioning, garbage collection, and many more. All these features are targeted at developing component-oriented software. Although you don't have the sheer power of C++, you become more productive faster. C# based on key points in the following sections:

* Simple
* Modern
* Object-oriented
* Type-safe
* Versionable
* Compatible
* Flexible

**2.3.4 ADO.NET – Database Connectivity**

ADO.NET provides a bridge between the front end controls and the back end database. The ADO.NET objects encapsulate all the data access operations and the controls interact with these objects to display data, thus hiding the details of movement of data.

ADO.NET is an object-oriented set of libraries that allows you to interact with data sources. Commonly, the data source is a database, but it could also be a text file, an Excel spreadsheet, or an XML file. For the purposes of this tutorial, we will look at ADO.NET as a way to interact with a data base.

As you are probably aware, there are many different types of databases available. For example, there is Microsoft SQL Server, Microsoft Access, Oracle, Borland Interbase, and IBM DB2, just to name a few.

We know that ADO.NET allows us to interact with different types of data sources and different types of databases. However, there isn't a single set of classes that allow you to accomplish this universally. Since different data sources expose different protocols, we need a way to communicate with the right data source using the right protocol Some older data sources use the ODBC protocol, many newer data sources use the OleDb protocol, and there are more data sources every day that allow you to communicate with them directly through .NET ADO.NET class libraries.

ADO.NET provides a relatively common way to interact with data sources, but comes in different sets of libraries for each way you can talk to a data source. These libraries are called Data Providers and are usually named for the protocol or data source type they allow you to interact with.

**2.4 Introduction to SQL Server:**

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

A database is one or more lists of values. A [computer database](http://www.functionx.com/sqlserver/Lesson01.htm) is one whose values are stored in a [computer medium](http://www.functionx.com/sqlserver/Lesson01.htm) such as a hard disk. A [desktop database](http://www.functionx.com/sqlserver/Lesson01.htm) is one that is used in one computer. A client/[server database](http://www.functionx.com/sqlserver/Lesson01.htm) is a database that is stored in one computer named a server and other computers named clients connect to the server to access and use the database. [Microsoft SQL Server](http://www.functionx.com/sqlserver/Lesson01.htm) is an application used to create computer databases for the Microsoft [Windows family](http://www.functionx.com/sqlserver/Lesson01.htm) of server operating systems. [Microsoft SQL](http://www.functionx.com/sqlserver/Lesson01.htm) Server provides an environment used to generate databases that can be accessed from workstations, the Internet, or other media such as a personal digital assistant (PDA).

Microsoft SQL Server is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](http://en.wikipedia.org/wiki/Concurrent_user). Its primary [query languages](http://en.wikipedia.org/wiki/Query_language) are [T-SQL](http://en.wikipedia.org/wiki/Transact-SQL) and [ANSI SQL](http://en.wikipedia.org/wiki/SQL).

**2.5 Internet Information Server (IIS)**

Internet Information Services (IIS, formerly Internet Information Server) is an extensible [web server](http://en.wikipedia.org/wiki/Web_server) created by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) for use with [Windows NT](http://en.wikipedia.org/wiki/Windows_NT) family. IIS supports [HTTP](http://en.wikipedia.org/wiki/HTTP), [HTTPS](http://en.wikipedia.org/wiki/HTTPS), [FTP](http://en.wikipedia.org/wiki/File_Transfer_Protocol), [FTPS](http://en.wikipedia.org/wiki/FTPS), [SMTP](http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) and [NNTP](http://en.wikipedia.org/wiki/Network_News_Transfer_Protocol). It has been an integral part of the Windows NT family since [Windows NT 4.0](http://en.wikipedia.org/wiki/Windows_NT_4.0), though it may be absent from some editions (e.g. Windows XP Home edition). IIS is not turned on by default when Windows is installed. The IIS Manager is accessed through the [Microsoft Management Console](http://en.wikipedia.org/wiki/Microsoft_Management_Console) or Administrative Tools in the Control Panel.

IIS 7.0 has a modular architecture. Modules, also called extensions, can be added or removed individually so that only modules required for specific functionality have to be installed. IIS 7 includes native modules as part of the full installation. These modules are individual features that the server uses to process requests and include the following:

* **Security modules**: Used to perform many tasks related to security in the request-processing pipeline, such as specifying authentication schemes, performing URL authorization, and filtering requests.
* **Content modules**: Used to perform tasks related to content in the request-processing pipeline, such as processing requests for static files, returning a default page when a client does not specify a resource in a request, and listing the contents of a directory.
* **Compression modules**: Used to perform tasks related to compression in the request-processing pipeline, such as compressing responses, applying Gzip compression transfer coding to responses, and performing pre-compression of static content.
* **Caching modules**: Used to perform tasks related to caching in the request-processing pipeline, such as storing processed information in memory on the server and using cached content in subsequent requests for the same resource.
* **Logging and Diagnostics modules**: Used to perform tasks related to logging and diagnostics in the request-processing pipeline, such as passing information and processing status to HTTP. Sys for logging, reporting events, and tracking requests currently executing in worker processes.

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 Introduction**

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In [software](http://searchsoa.techtarget.com/definition/software) engineering, such requirements are often called [functional specification](http://searchsoftwarequality.techtarget.com/definition/functional-specification)s. Requirements analysis is an important aspect of [project management](http://searchcio-midmarket.techtarget.com/definition/project-management). Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of [feature creep](http://searchcio.techtarget.com/definition/feature-creep) and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements. Requirements analysis is a team effort that demands a combination of [hardware](http://searchcio-midmarket.techtarget.com/definition/hardware), software and [human factors](http://searchsoa.techtarget.com/definition/human-factors) engineering expertise as well as skills in dealing with people.

**3.2 Non Functional Requirements**

**3.2.1 Hardware Interface**

Hardware required to develop the software is as listed below

* Processor : Pentium IV onwards
* RAM : 2GB +
* Hard disk space : 40GB +
* Standard PC configuration to carryout challenging computing

**3.2.2 Software Interface**

Software required in development is as listed below

Operating System : Windows XP or Higher

IDE : Visual Studio 2010

Language : C#

Framework : ASP.NET 4.0

Back End : MS SQL Server

**3.2.3 Communication Interface**

**HTTP Protocol:**

The Hypertext Transfer Protocol (HTTP) is an [application protocol](http://en.wikipedia.org/wiki/Application_protocol) for distributed, collaborative, [hypermedia](http://en.wikipedia.org/wiki/Hypermedia) information systems. HTTP is the foundation of data communication for the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web). [Hypertext](http://en.wikipedia.org/wiki/Hypertext) is structured text that uses logical links ([hyperlinks](http://en.wikipedia.org/wiki/Hyperlinks)) between [nodes](http://en.wikipedia.org/wiki/Node_(computer_science)) containing text. HTTP is the protocol to exchange or transfer hypertext

## HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet).

HTTP concepts include (as the Hypertext part of the name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any [Web server](http://whatis.techtarget.com/definition/Web-server) machine contains, in addition to the Web page files it can serve, an HTTP [daemon](http://searchsoa.techtarget.com/definition/daemon), a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web [browser](http://searchwindevelopment.techtarget.com/definition/browser) is an HTTP [client](http://searchenterprisedesktop.techtarget.com/definition/client), sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator or [URL](http://searchnetworking.techtarget.com/definition/URL)) or clicking on a [hypertext](http://searchsoa.techtarget.com/definition/hypertext) [link](http://searchsoa.techtarget.com/definition/link), the browser builds an HTTP request and sends it to the Internet Protocol address ([IP address](http://searchwindevelopment.techtarget.com/definition/IP-address)) indicated by the URL. The HTTP daemon in the destination server machine receives the request and sends back the requested file or files associated with the request. (A Web page often consists of more than one file.)

**3.3 Functional Requirements**

**System Functions**

There are five actors in this project. They are:

1. Admin
2. Users
3. Chef
4. Farmer
5. Retailer

**1) Admin:-**

* Create Food Type
* Create Food
* Create Chef
* Food Availability
* Chef Orders
* Products List
* Report

**2) Chef:-**

* Prepare Food
* Order Product
* Profile Update
* Password Update

**3) User:-**

* Browse Food
* Cart
* Profile Update
* Password Update

**4) Farmer:-**

* Manage Products
* Profile Update
* Password Update

**5) Retailer:-**

* Manage Products
* Profile Update
* Password Update

**CHAPTER 4**

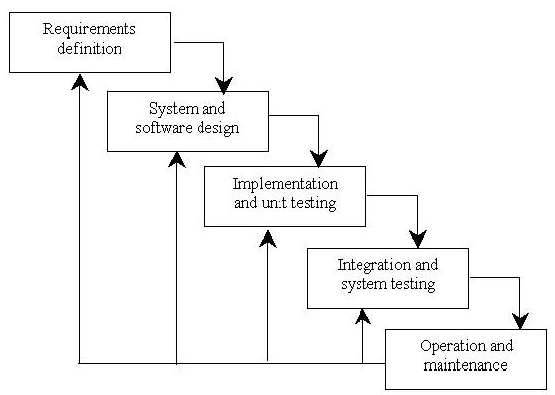
**SYSTEM ANALYSIS AND DESIGN**

**4.1 Approach Adapted**

**Waterfall Model**

Waterfall model is an example of Sequential model. In this model, the software development activity is divided into different phases and each phase consists of series of tasks and has different objectives. Waterfall model is the pioneer of the [SDLC processes](http://en.wikipedia.org/wiki/Software_development_process). In fact it was the first model which was widely used in the software industry. It is divided into phases and output of one phase becomes input of the next phase. It is mandatory for a phase to be completed before the next phase starts. In short, there is no overlapping in Waterfall model

In waterfall, development of one phase starts only when the previous phase is complete. Because of this nature, each phase of waterfall model is quite precise well defined. Since the phase’s falls from higher level to lower level, like a water fall, it’s named as waterfall model.



**Figure 4.1: Waterfall model**

* **Requirement Analysis & Definition:** This phase is focused on possible requirements of the system for the development are captured. Requirements are gathered subsequent to the end user consultation.
* **System & Software Design:** Prior to beginning the actual coding, it is inevitable to understand what actions are to be taken and what they should like. The requirement specifications are studied in detail in this phase and the design of the system is prepared. The design specifications are the base for the implementation and unit testing model phase.
* **Implementation & Unit Testing:** Subsequent to receiving the system design documents, the work is shared into various modules and the real coding is commenced. The system is developed into small coding units. These units are later integrated in the subsequent phase.

Every unit is tested for its functionality.

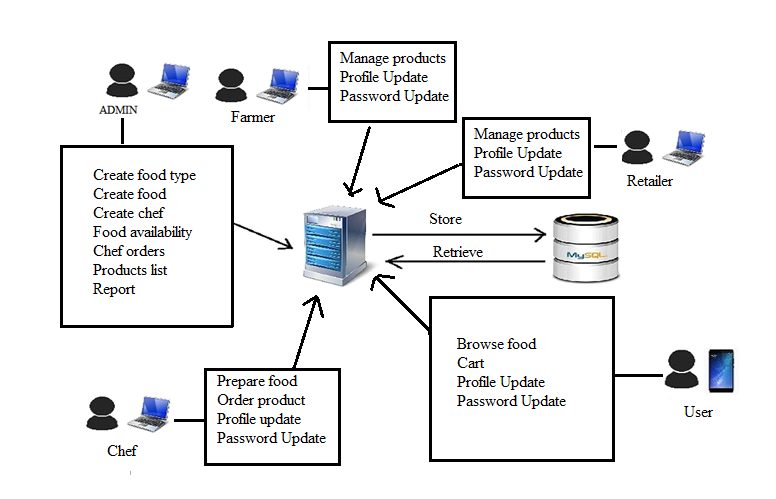
* **Integration & System Testing:** The modules that are divided into units are integrated into a complete system and tested for proper coordination among modules and system behaves as per the specifications. Once the testing is completed, the software product is delivered to the customer.
* **Operations & Maintenance:** It is a never ending phase. Once the system is running in production environment, problems come up. The issues that are related to the system are solved only after deployment of the system. The problems arise from time to time and need to be solved; hence this phase is referred as maintenance. Unlike the waterfall model, the phased model is suitable if the work can be grouped into separate units and delivered in steps rather than everything once and together, by different teams. Consider a system that consists of 4 subsystems, each being developed by a separate team. In the end all the 4 subsystems make up one complete system, giving the flexibility of breaking the system down in 4 parts and allowing each being developed separately. It’s more like a collection of mini projects run by different teams approach

**4.2 System Design**

Systems design is the process of defining the [architecture](http://en.wikipedia.org/wiki/Systems_architecture), components, modules, interfaces, and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). Systems design could be seen as the application of [systems theory](http://en.wikipedia.org/wiki/Systems_theory) to [product development](http://en.wikipedia.org/wiki/Product_development). There is some overlap with the disciplines of [systems analysis](http://en.wikipedia.org/wiki/Systems_analysis), [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering). The architectural design of a system emphasizes on the design of the [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) which describes the [structure](http://en.wikipedia.org/wiki/Structure), [behaviour](http://en.wikipedia.org/wiki/Behavior), and more [views](http://en.wikipedia.org/wiki/View_model) of that system.

**4.2.1 Architectural Design**

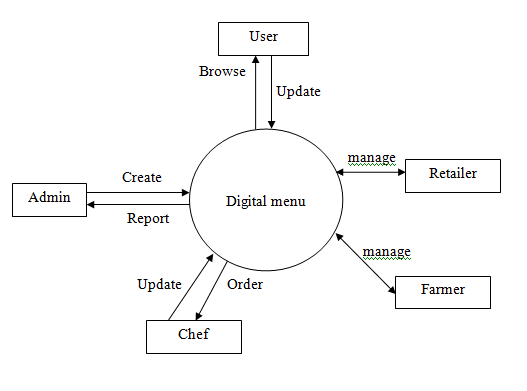
Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on identifying components or subsystems and how they connect. In other words, the focus is on what major components are needed.

A

**Figure 4.2: Architecture of Digital Table Menu**

**4.2.2 Context data flow diagram:**

It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the 'Level 0 DFD') the system's interactions with the outside world are modelled purely in terms of data flows across the *system boundary*. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



**Figure 4.3: DFD for Admin**

**4.2.3 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system). DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a [flowchart](http://en.wikipedia.org/wiki/Flowchart), which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

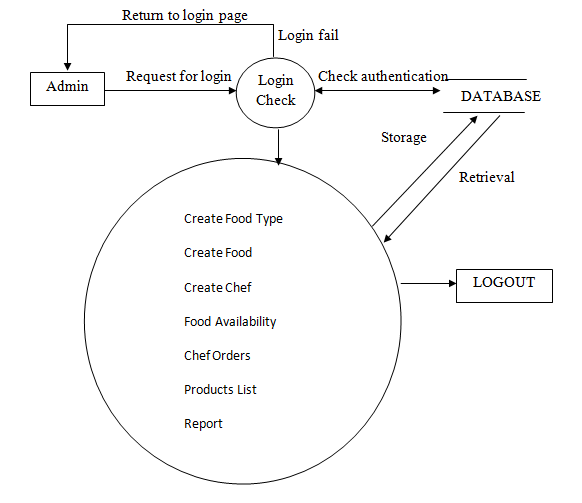
**Symbols used in DFD’s:**

**Processes:**  A process transforms data values. The lowest processes are our functions without side effects.

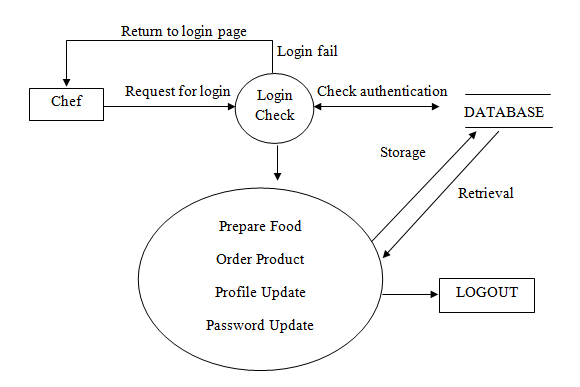
**Data Flows:**  A data flow connects the output of an object or process to the input of another object or process. It represents the intermediate data values within the computation. It is draws as an arrow between the procedure and the consumer of the data value. The arrow is labelled with the description of the data, usually its name or type.

**Actors:**  An actor is an active object that drives the data flow graph by producing or consuming values. Actors are attached to the inputs and the outputs of a dataflow graph. In sense, the actors lie on the boundary of the flow graph but terminate the flow of data as sources and sinks of data, and so are sometimes called terminators.

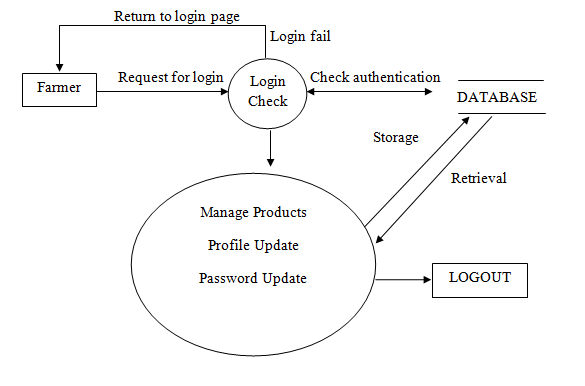
**Data Store:**  A data store is a passive object within a data flow diagram that stores data for later access. Unlike an actor, a data store does not generate any operations on its own but merely responds to requests to store and access data.

****

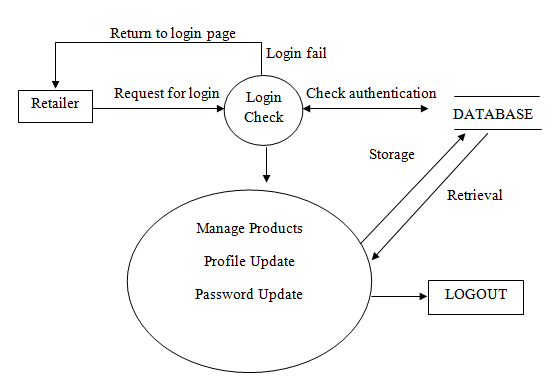
**Figure 4.4: DFD for Admin Owner**

****

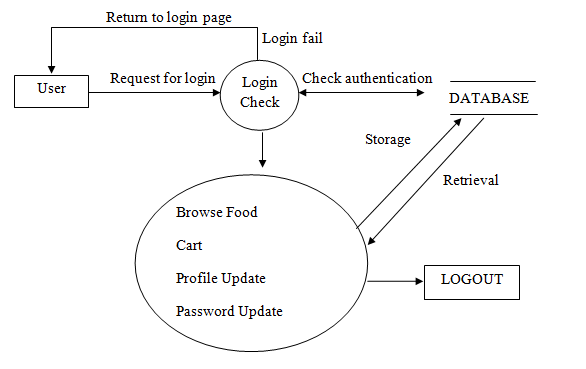
**Figure 4.5: DFD for Chef**

****

**Figure 4.6: DFD for Farmer**

****

**Figure 4.7: DFD for Retailer**

****

**Figure 4.8: DFD for User**

**4.2.4 Use Case Diagram**

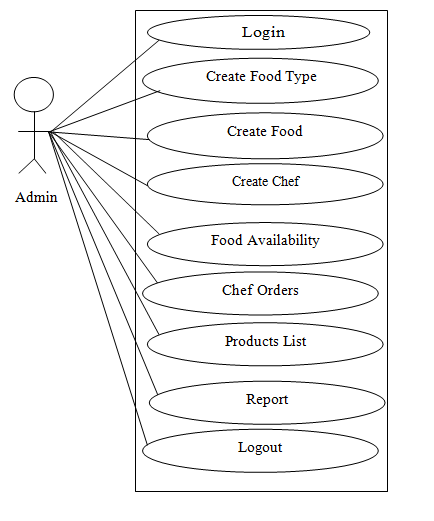
Use case diagrams are considered for high level requirement analysis of a system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. Now when the initial task is complete use case diagrams are modeled to present the outside view.

**Use case:**

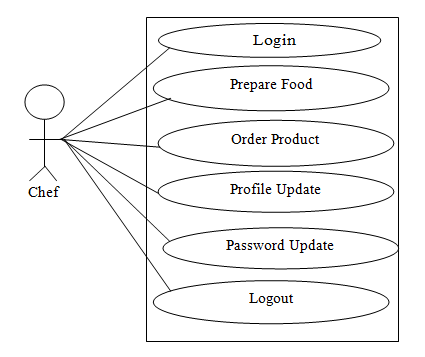
Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say that uses cases are nothing but the system functionalities written in an organized manner.

**Actor:**

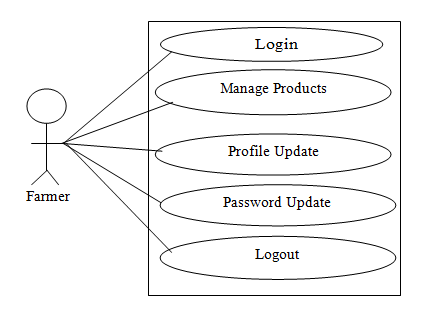
Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications.



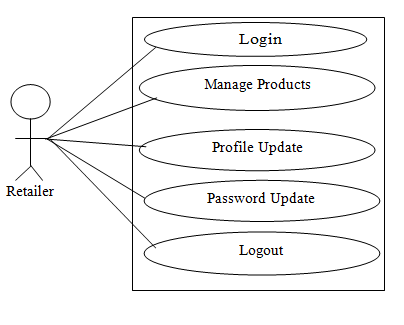
**Figure 4.9: Use Case for Admin**



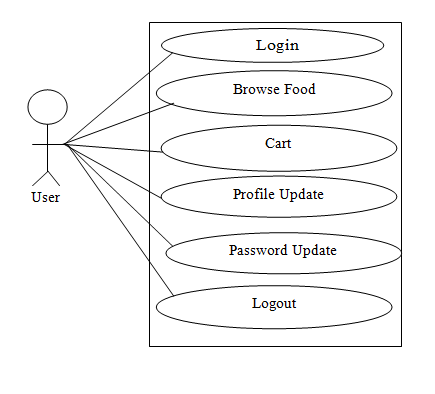
**Figure 4.10: Use Case for Chef**



**Figure 4.11: Use Case for Farmer**



**Figure 4.12: Use Case for Retailer**



**Figure 4.13: Use Case for User**

**4.2.5 Sequence Diagram**

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It describe interactions among classes in terms of an exchange of messages over time. Sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom

**Targets/Class roles/State:**

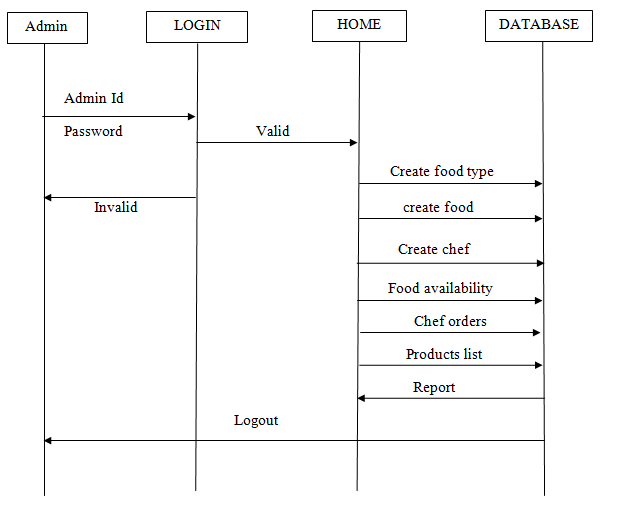
Objects as well as classes can be targets on a sequence diagram, which means that messages can be sent to them. A target is displayed as a rectangle with some text in it. Below the target, its lifeline extends for as long as the target exists. Targets can be actor, boundary, control, entity and database.

**Messages:**

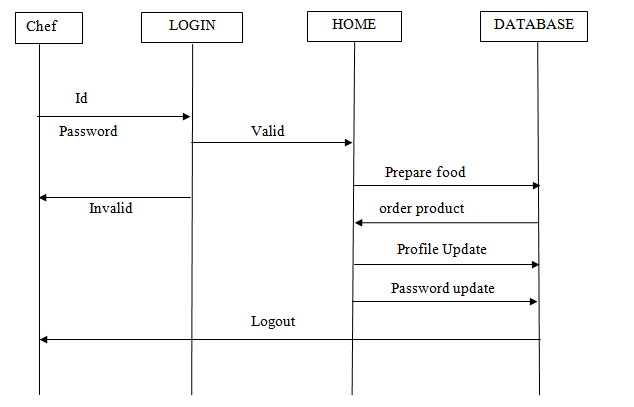
Messages are arrows that represent communication between objects.

**Lifelines:**

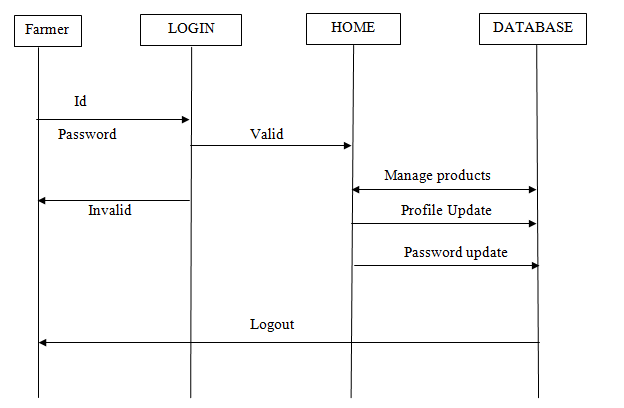
Lifelines are vertical dashed lines that indicate the object's presence over time.

****

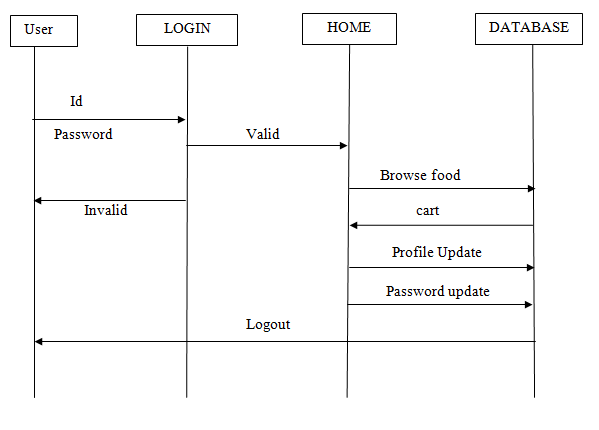
**Figure 4.14: Sequence Diagram for Admin**

****

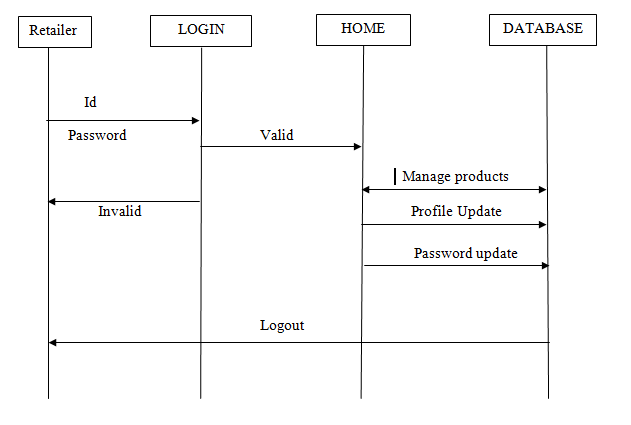
**Figure 4.15: Sequence Diagram for Chef**

****

**Figure 4.16: Sequence Diagram for Farmer**

****

**Figure 4.17: Sequence Diagram for User**

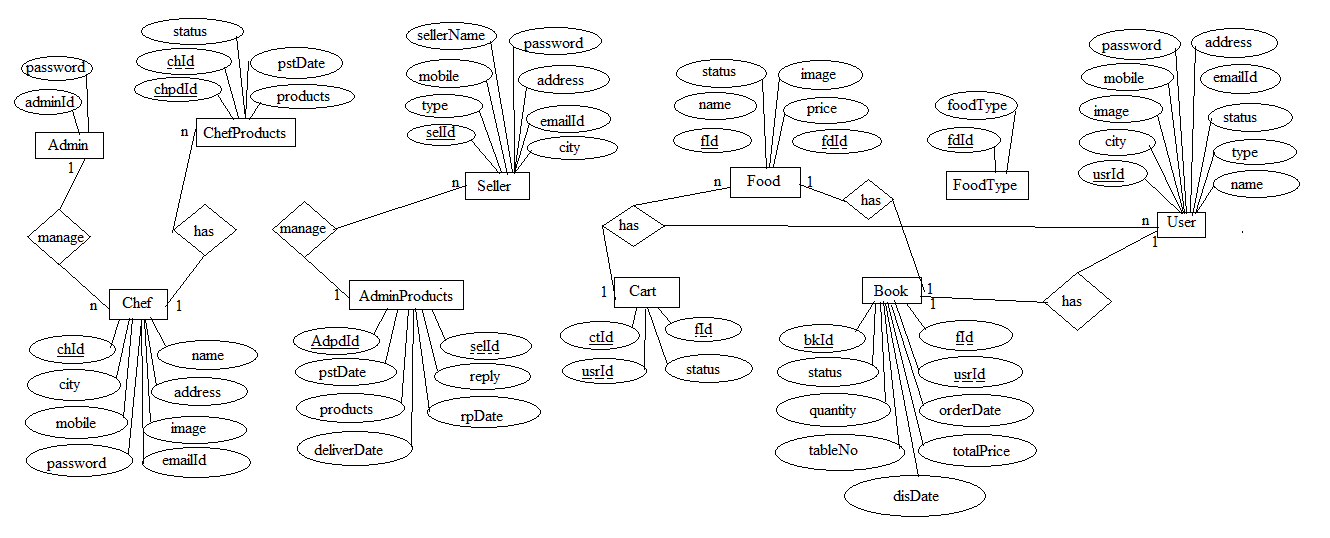
****

**Figure 4.18: Sequence Diagram for Retailer**

**4.2.6 Entity Relationship Diagram**

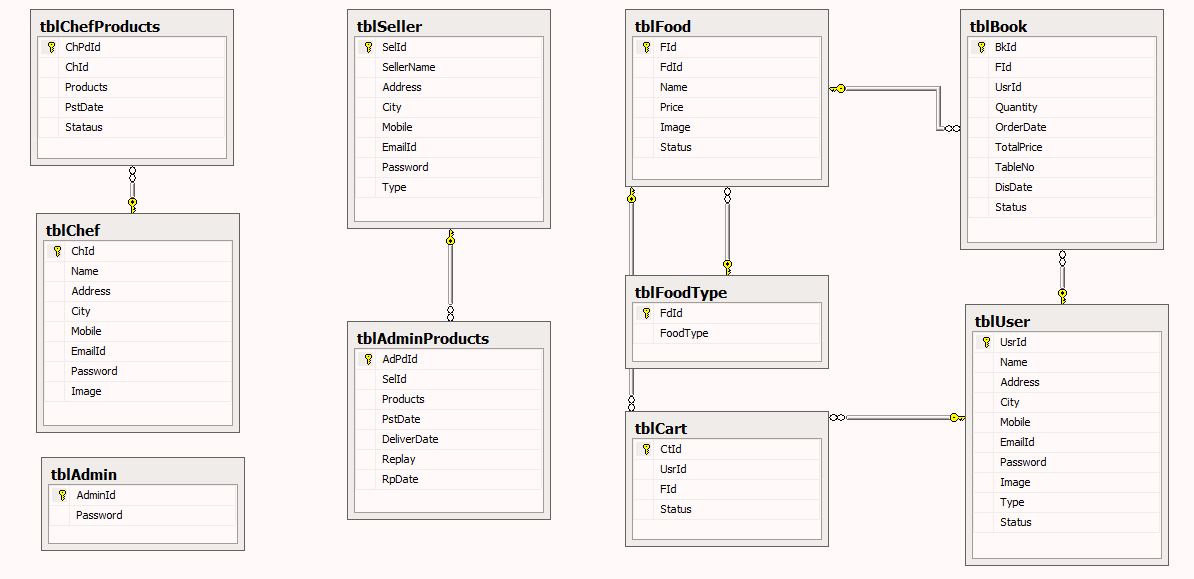
Entity–relationship model(ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a [relational database](http://en.wikipedia.org/wiki/Relational_database). The main components of ER models are [entities](http://en.wikipedia.org/wiki/Entities) (things) and the relationships that can exist among them, and databases. ER Diagram is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored. A relationship is how the data is shared between entities. There are three types of relationships between entities:

* **One-to-One:** One instance of an entity is associated with one other instance of another entity.
* **One-to-Many:** One instance of an entity is associated with zero, one or many instances of another entity, but for one instance of entity B there is only one instance of entity A.
* **Many-to-Many**: One instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A.



**Figure 4.19: ER Diagram**

**4.2.7 Relations Established in Database**

The Database design will represent in ER diagram. The diagram contains the overall relationship of tables and detailed design of the database. This diagram helps us to now the design of project database. In this project the six entities are present entities are like a database tables. 

**Figure 4.20: DB diagram**

**CHAPTER 5**

**IMPLEMENTATION**

This web application is implemented using object oriented programming language. Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

**Features of Object Oriented paradigm:**

* Emphasis is on data rather than procedure.
* Programs are divided into what are known as objects.
* Data structures are designed such that they characterize the objects.
* Methods that operate on the data of an object are tied together in the data structure.
* Objects may communicate with each other through methods.
* New data and methods can be easily added whenever necessary.
* Follows bottom-up approach in program design.
* Data is hidden and cannot be accessed by external functions.

This project is implemented using n tier architecture. ASP.NET is used in the presentation layer, C# classes are used in the Business logic, Table adopter is used in the data tier and MS SQL server 2005 (database) is used as the backend.

**Code of** **Admin:**

namespace DigitalMenu.Controllers

{

public ActionResult Login(string returnUrl)

{

ViewBag.ReturnUrl = returnUrl;

return View();

}

//

// POST: /Account/Login

[HttpPost]

[AllowAnonymous]

[ValidateAntiForgeryToken]

public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)

{

if (!ModelState.IsValid)

{

return View(model);

}

// This doesn't count login failures towards account lockout

// To enable password failures to trigger account lockout, change to shouldLockout: true

var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, shouldLockout: false);

switch (result)

{

case SignInStatus.Success:

return RedirectToLocal(returnUrl);

case SignInStatus.LockedOut:

return View("Lockout");

case SignInStatus.RequiresVerification:

return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });

case SignInStatus.Failure:

default:

ModelState.AddModelError("", "Invalid login attempt.");

return View(model);

}

}

**------------------------------------------------------------------------------------------------**

**Code of** **Chef:**

namespace DigitalMenu.Controllers

{

public class ChefController : Controller

{

DigitalMenuEntities1 obj = new DigitalMenuEntities1();

// GET: Chef

public ActionResult ChefHome()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

return View();

}

public ActionResult ChefProducts()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

ViewBag.Res = (from c in obj.tblChefProducts select c).ToList();

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult ChefProducts(tblChefProduct cp)

{

string em = Session["User"].ToString();

int cId = (from c in obj.tblChefs where c.EmailId == em select c.ChId).SingleOrDefault();

cp.ChId = cId;

cp.PstDate = DateTime.Now.ToShortDateString();

cp.Stataus = "Pending";

//var dateAsString = DateTime.Now.ToString("yyyy-MM-dd");

obj.tblChefProducts.Add(cp);

obj.SaveChanges();

TempData["InsertChPro"] = "Products List Inserted Sucessfully!";

return RedirectToAction(nameof(ChefProducts));

}

public ActionResult ChefPrepFood()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

ViewBag.Res = obj.tblBooks.Where(d => d.Status == "Pending").Include(d => d.tblFood).ToList();

return View();

}

[HttpPost]

public ActionResult ChefPrepFood(string id)

{

int ids = int.Parse(id);

tblBook bk = obj.tblBooks.Where(x => x.BkId == ids).FirstOrDefault();

bk.DisDate = DateTime.Now;

bk.Status = "Finished";

obj.SaveChanges();

return RedirectToAction(nameof(ChefPrepFood));

}

**------------------------------------------------------------------------------------------------**

**Code of** **User:**

namespace DigitalMenu.Controllers

{

public class UserController : Controller

{

DigitalMenuEntities1 obj = new DigitalMenuEntities1();

// GET: User

public ActionResult UserHome()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

return View();

}

public ActionResult Menu()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

ViewBag.FT = new SelectList(obj.tblFoodTypes, "FdId", "FoodType");

if (Session["FoodTp"] != null)

{

try

{

int id = int.Parse(Session["FoodTp"].ToString());

int tp = int.Parse(TempData["FD"].ToString());

ViewBag.Res = (from f in obj.tblFoods where f.FdId == id && f.Status == "Available" select f).ToList();

}

catch

{

}

}

return View();

}

[HttpPost]

public ActionResult Menu(string st)

{

string tpId = Request.Form["type"];

TempData["FD"] = tpId;

Session["FoodTp"] = tpId;

return RedirectToAction(nameof(Menu));

}

public ActionResult GetFdId(int id)

{

TempData["FdID"] = id;

string em = Session["User"].ToString();

if (id != 0)

{

tblCart crt = new tblCart();

int uID = (from u in obj.tblUsers where u.EmailId == em select u.UsrId).SingleOrDefault();

crt.UsrId = uID;

crt.FId = id;

crt.Status = "Cart";

obj.tblCarts.Add(crt);

obj.SaveChanges();

TempData["AddCart"] = "Food Added to Cart!";

}

return RedirectToAction(nameof(Menu));

}

**------------------------------------------------------------------------------------------------**

**Code of** **Farmer:**

namespace DigitalMenu.Controllers

{

public class FormerController : Controller

{

DigitalMenuEntities1 obj = new DigitalMenuEntities1();

// GET: Former

public ActionResult FormerHome()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

return View();

}

public ActionResult GetProdsList()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

string em = Session["User"].ToString();

int id = (from s in obj.tblSellers where s.EmailId == em select s.SelId).SingleOrDefault();

var res = (from p in obj.tblAdminProducts where p.SelId == id select p).ToList();

ViewBag.Res = res;

return View();

}

public ActionResult GetProID(int id)

{

//ScriptManager.RegisterStartupScript(this, this.GetType(), "Pop", "$(document).ready(function() {$('#myModal').modal('show');});", true);

Session["AdProID"] = id;

return Json(null, JsonRequestBehavior.AllowGet);

}

public ActionResult ActProductList()

{

int id = int.Parse(Session["AdProID"].ToString());

string rep = Request.Form["rep"];

tblAdminProduct ad = obj.tblAdminProducts.Where(x => x.AdPdId == id).SingleOrDefault();

var dt = DateTime.Now.ToShortDateString();

ad.Replay = rep;

ad.RpDate = dt;

obj.SaveChanges();

return RedirectToAction(nameof(GetProdsList));

}

**------------------------------------------------------------------------------------------------**

**Code of** **Seller:**

namespace DigitalMenu.Controllers

{

public class SellerController : Controller

{

DigitalMenuEntities1 obj = new DigitalMenuEntities1();

// GET: Retailer

public ActionResult RetailerHome()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

return View();

}

public ActionResult GetProdList()

{

if (Session["User"] == null)

return RedirectToAction("Login", "Home");

string em = Session["User"].ToString();

int id = (from s in obj.tblSellers where s.EmailId == em select s.SelId).SingleOrDefault();

var res = (from p in obj.tblAdminProducts where p.SelId == id select p).ToList();

ViewBag.Res = res;

return View();

}

**CHAPTER 6**

**TESTING**

Testing is a process of executing a program to ensure that defined input will produce actual results that agree with required outputs. In developing a software project, error can be initiated at any stage during the development. For each phase of the software development cycle there are different techniques for detecting and elimination errors that originate in that phase. However some errors will reflect in the code. Testing performs a very crucial role for quality assurance and for ensuring the reliabilities of the software. The quality of the system depends on its design, development, testing and implementation. Weaknesses in any of these areas will seriously affect the quality and therefore value of the system to its users. Once the code has been generated, testing of the modules begins implementation ends with formal tests.

**6.1** **Purpose of Testing:**

Testing accomplishes a variety of things, but most importantly it measures the quality of the software we are developing. This view presupposes there are defects in the software waiting to be discovered and this view is rarely disproved or even disputed.

Several factors contribute to the importance of making testing a high priority of any software development effort. These include:

* Reducing the cost of developing the program.
* Ensuring that the application behaves exactly as we explain to the user for the vast majority of programs, unpredictability is the least desirable consequences of using an application.
* Reducing the total cost of ownership. By providing software that looks and behaves as shown in the documentation, the customers require fewer hours of training and less support from product experts.
* Developing customer loyalty and word-of-mouth market share.

**6.2 Different Types of Testing:**

**6.2.1 Unit Testing:**

Unit testing focuses verification on the smallest unit of software design, the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The unit testing is a white box oriented testing.

First of all the module interface is tested to ensure that the information properly flows into and out of the program until under test. Then the local data structure is tested to ensure the data stored temporarily maintains its integrity during all steps in an execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once. And finally, all errors handling paths are tested. In this project the testing is done according to bottom-up approach. Starting with smallest and lowest level modules and processing one at a time. For each module a driver and corresponding stubs were also written. If any errors found they were corrected immediately and the unit was tested again.

**6.2.2 Integration Testing:**

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process are tested together. Any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis.

In this software, the bottom-up integration testing approached has been used, starting with the smallest and lowest level modules and proceeding one at a time. For each module the tests were conducted and the results were noted down.

**6.2.3 User Testing:**

User Testing is nothing but the test of the software by the users themselves with live data being fed to the system. This helps in building really robust system. User testing in this system has been done extensively ascertain the results.

**6.2.4 White Box Testing**

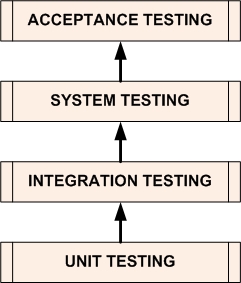
White box software testing is the testing of the working of the software and its internal structures. It can detect errors of the implemented parts, but the unimplemented parts goes undetected. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning. For white box testing, the test cases cannot be determined until the code has actually been written. Test data are derived from direct examination of the code to be tested.

**6.2.4 Black Box Testing**

Black box testing is the testing of the functionality of the software as opposed to its internal structure. It can be done at all levels of software testing. Black Box Testing is testing without knowledge of the internal workings of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the “legal” inputs and what the expected outputs should be, but not how the program actually arrives at those outputs.

In black box testing, the test planning can begin as soon as the specifications are written. The opposite of this would be glass box testing, where test data are derived from direct examination of the code to be tested. Both of these testing techniques have advantages and disadvantages, but when combined, they help to ensure thorough testing of the product.

**6.3 levels of Testing**



**Figure 6.1: Testing Levels**

**[1. Unit Testing](http://softwaretestingfundamentals.com/unit-testing/)**

[Unit Testing](http://softwaretestingfundamentals.com/unit-testing/) is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

**[2. Integration Testing](http://softwaretestingfundamentals.com/integration-testing/)**

[Integration Testing](http://softwaretestingfundamentals.com/integration-testing/) is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

**3.** [**System Testing**](http://softwaretestingfundamentals.com/system-testing/)

[System Testing](http://softwaretestingfundamentals.com/system-testing/) is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

**[4.](http://softwaretestingfundamentals.com/acceptance-testing/)** [**[Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/)**](http://softwaretestingfundamentals.com/acceptance-testing/)

[Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/) is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

**5. Regression Testing**

Regression testing is a type of [software testing](http://en.wikipedia.org/wiki/Software_testing) that seeks to uncover new [software bugs](http://en.wikipedia.org/wiki/Software_bug), or [regressions](http://en.wikipedia.org/wiki/Software_regression), in existing [functional](http://en.wikipedia.org/wiki/Functional_testing) and [non-functional](http://en.wikipedia.org/wiki/Non-functional_testing) areas of a system after changes such as enhancements, [patches](http://en.wikipedia.org/wiki/Patch_(computing)) or [configuration](http://en.wikipedia.org/wiki/Configuration_file) changes, have been made to them.

The intent of regression testing is to ensure that changes such as those mentioned above have not introduced new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software.

Common methods of regression testing include rerunning previously completed tests and checking whether program behavior has changed and whether previously fixed faults have re-emerged. Regression testing can be performed to test a system efficiently by systematically selecting the appropriate minimum set of tests needed to adequately cover a particular change.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

**6.4 Test cases**

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program

Path or to verify compliance with a specific requirement. Test cases underline testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desire coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

**6.4.1 Test Scenario 1:** Enter proper credentials, login as Admin and check for successful login.

**Test Case 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Description | Input | Expected result | Actual result | Status |
| 1 | Open the application | N/A | ‘Digital Menu’ home page must be displayed | ‘Digital Menu’ ‘home page is displayed | Pass |
| 2 | Click on login | N/A | Login form must be displayed | Login form is displayed | Pass |
| 3 | Enter user Id and password,  Click on login | user Id: Admin  Password: 123 | Admin home page must be displayed | Admin home page is displayed | Pass |

**Table 6.1:** Test case for Admin

**6.4.2 Test Scenario 2:** open mobile application, register as user.

**Test Case 2:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Description | Input | Expected result | Actual result | | Status | |
| 1 | Open the mobile application | N/A | Login screen must be displayed for registration | Login screen is displayed for registration | | Pass | |
| 2 | click on register | N/A | ‘registration’ screen  must be displayed | | ‘registration’  screen is  displayed | | Pass |
| 3 | Enter proper details for registration  Click on register |  | ‘User registered successfully’ message must be displayed. | | ‘User registered successfully’ message is displayed. | | Pass |

**Table 6.2:** Test case for User

**6.4.3 Test Scenario 3:** Enter improper credentials, login as Admin and check for error message.

**Test Case 3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Description | Input | Expected result | Actual result | Status |
| 1 | Open the application | N/A | ‘Digital Menu’ home page must be displayed | ‘Digital Menu’ home page is displayed | Pass |
| 2 | Click on login | N/A | Login form must be displayed | Login form is displayed | Pass |
| 3 | Enter user Id and password,  Click on login | user Id: Admin1  Password: 123 | ‘invalid username/password’ message must be displayed | ‘invalid username/password’ message is displayed | Pass |

**Table 6.3:** Test case for Error message by Admin

**6.4.4 Test Scenario 4:** Enter improper credentials, login as user in the mobile application and check for error message.

**Test Case 4:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Description | Input | Expected result | Actual result | Status |
| 1 | Open the application | N/A | ‘menu’ screen must be displayed | ‘menu’ screen is displayed | Pass |
| 2 | Enter mobile number and password,  Click on login | Mobile  9978605654  Password  564d#w | |  | | --- | | ‘Already Registered’ message must be displayed. | | |  | | --- | | ‘Already registered ‘ message is displayed | | Pass |

**Table 6.4:** Test case for Error message by User

**CHAPTER 7**

**SNAPSHOTS**

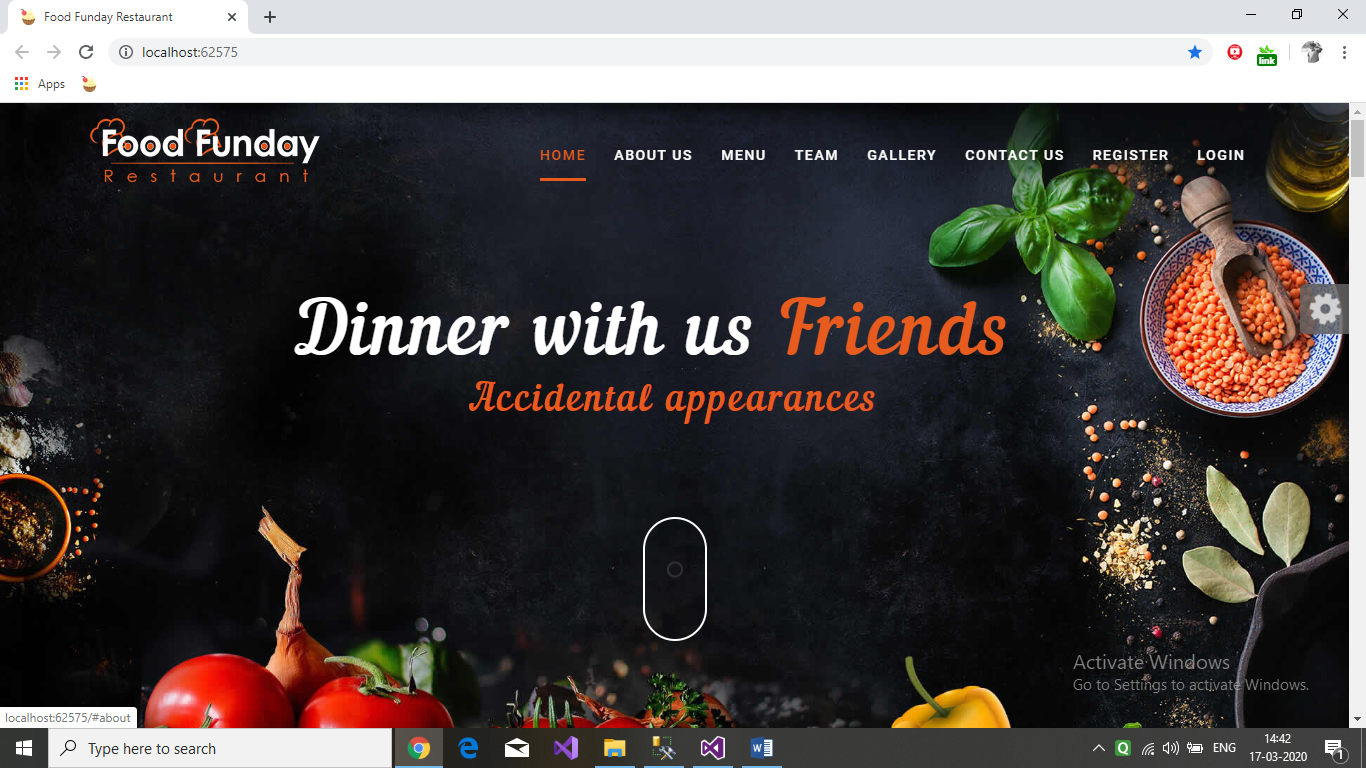


Figure 7.1: **Home Page**

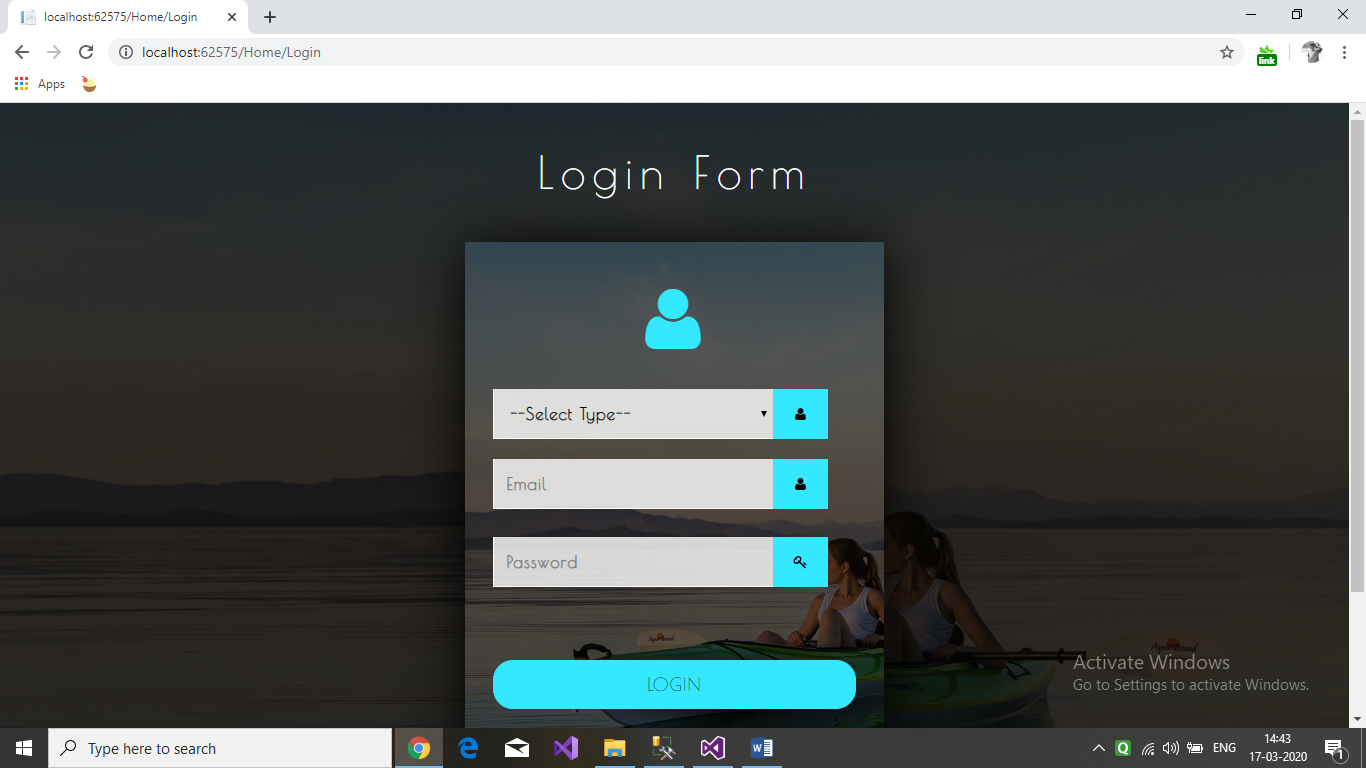


Figure 7.2: **Login Page**

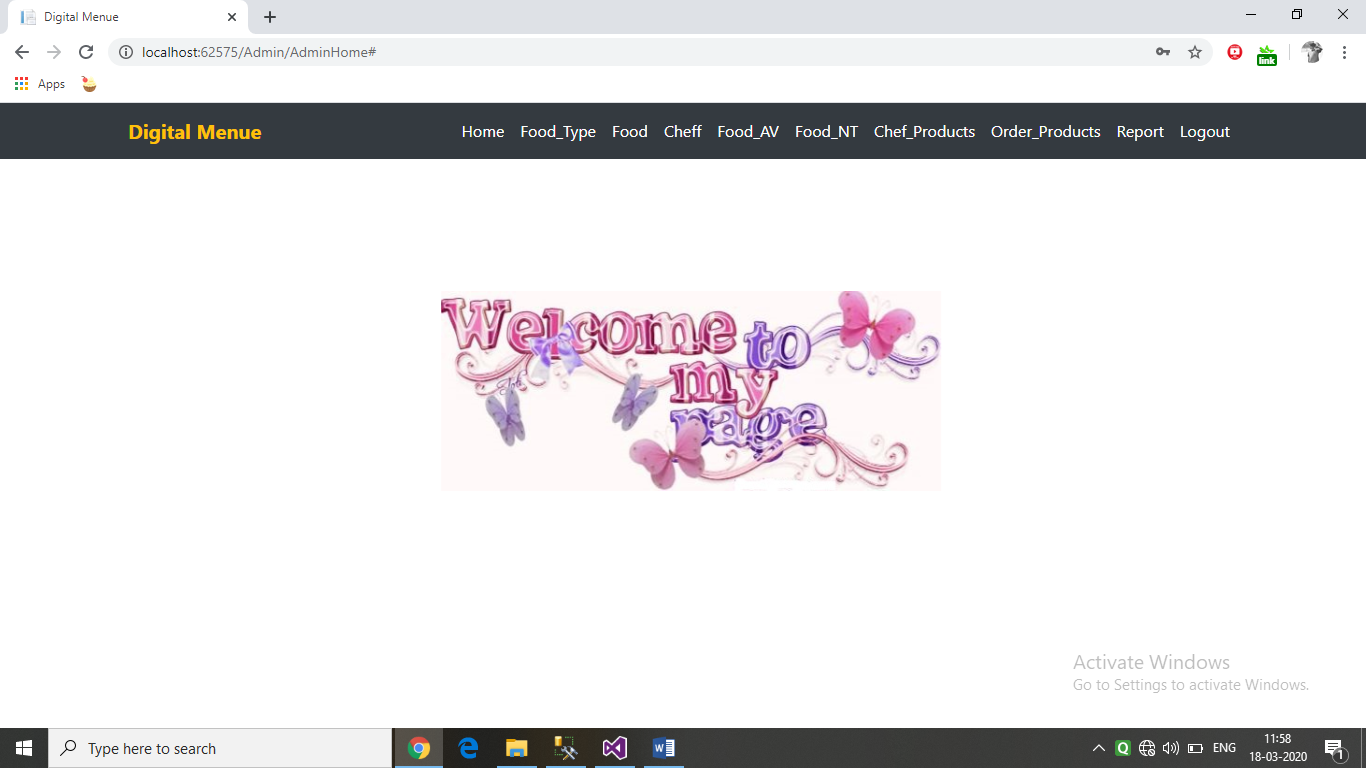


Figure 7.3: **Welcome Page**

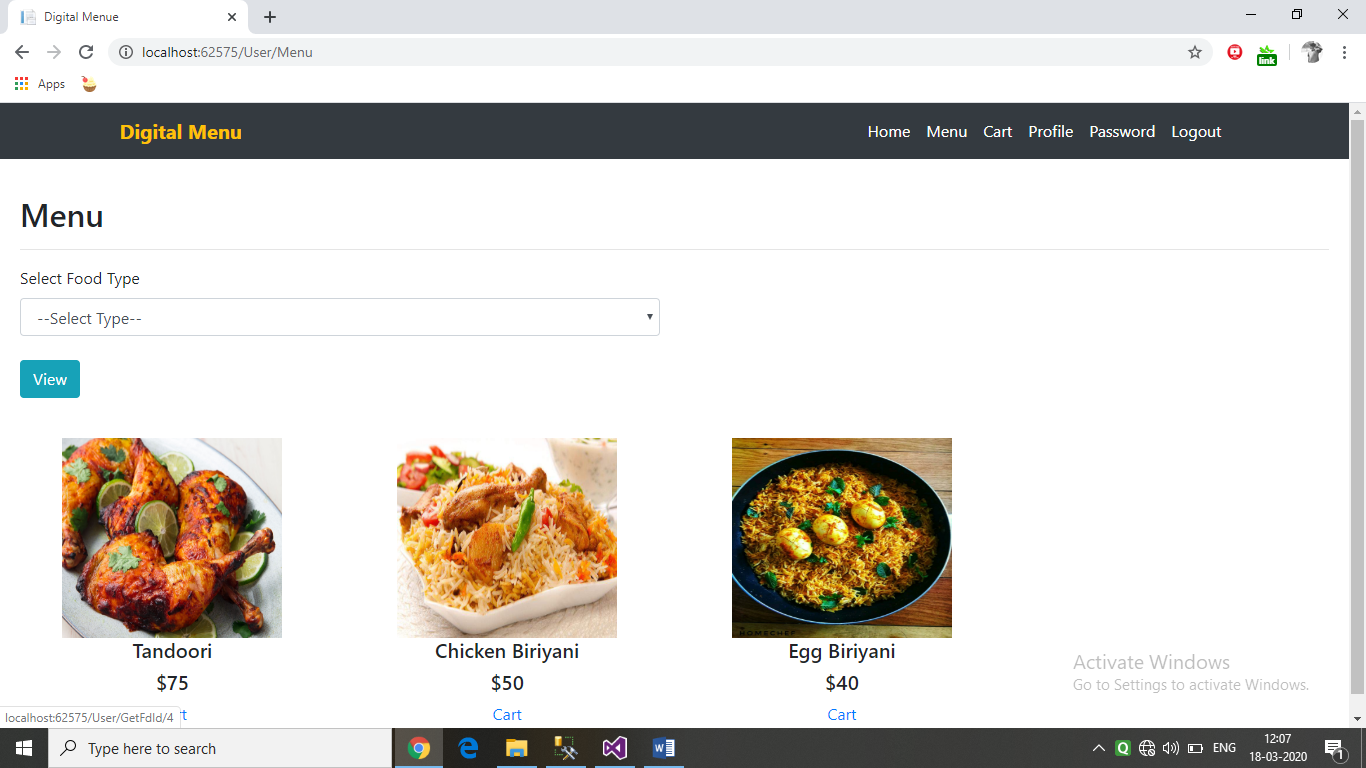


Figure 7.4: **Menu Page**

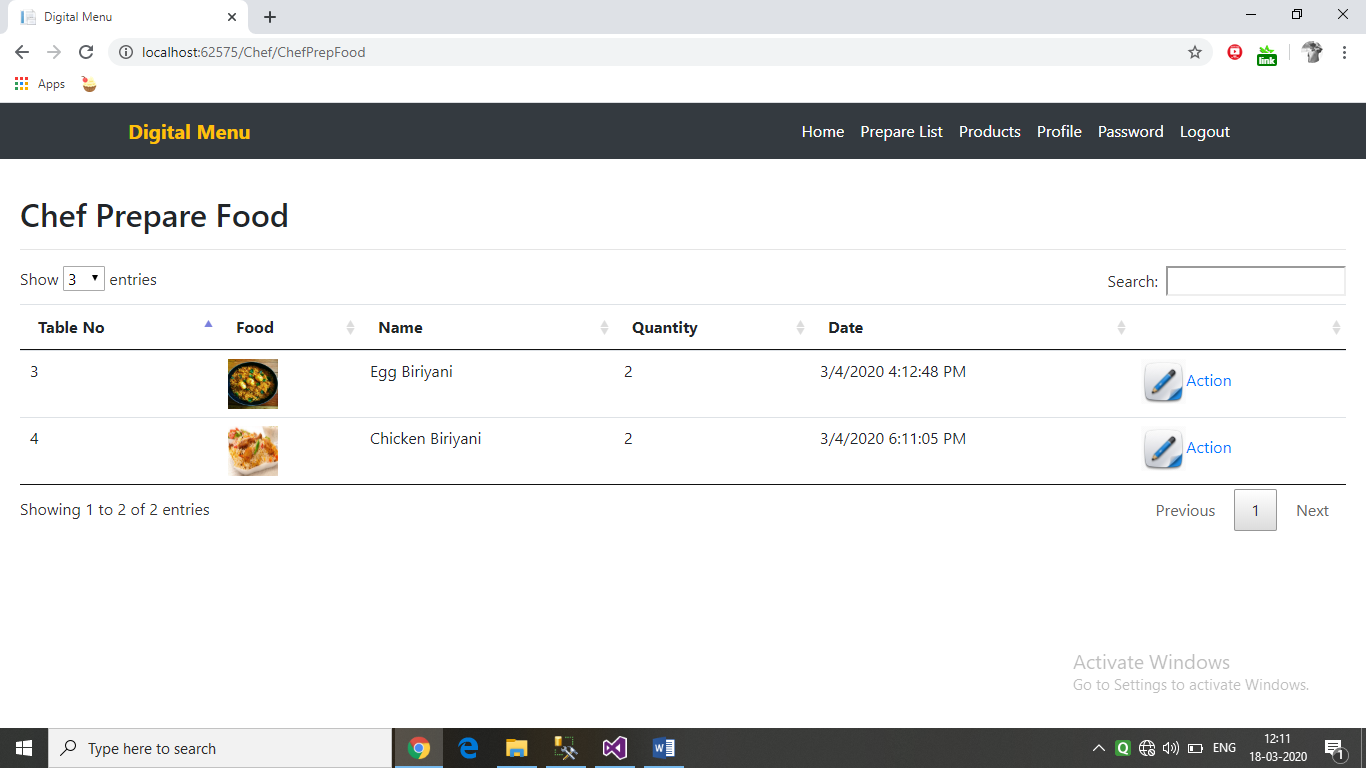


Figure 7.5: **Chef’s food status Page**

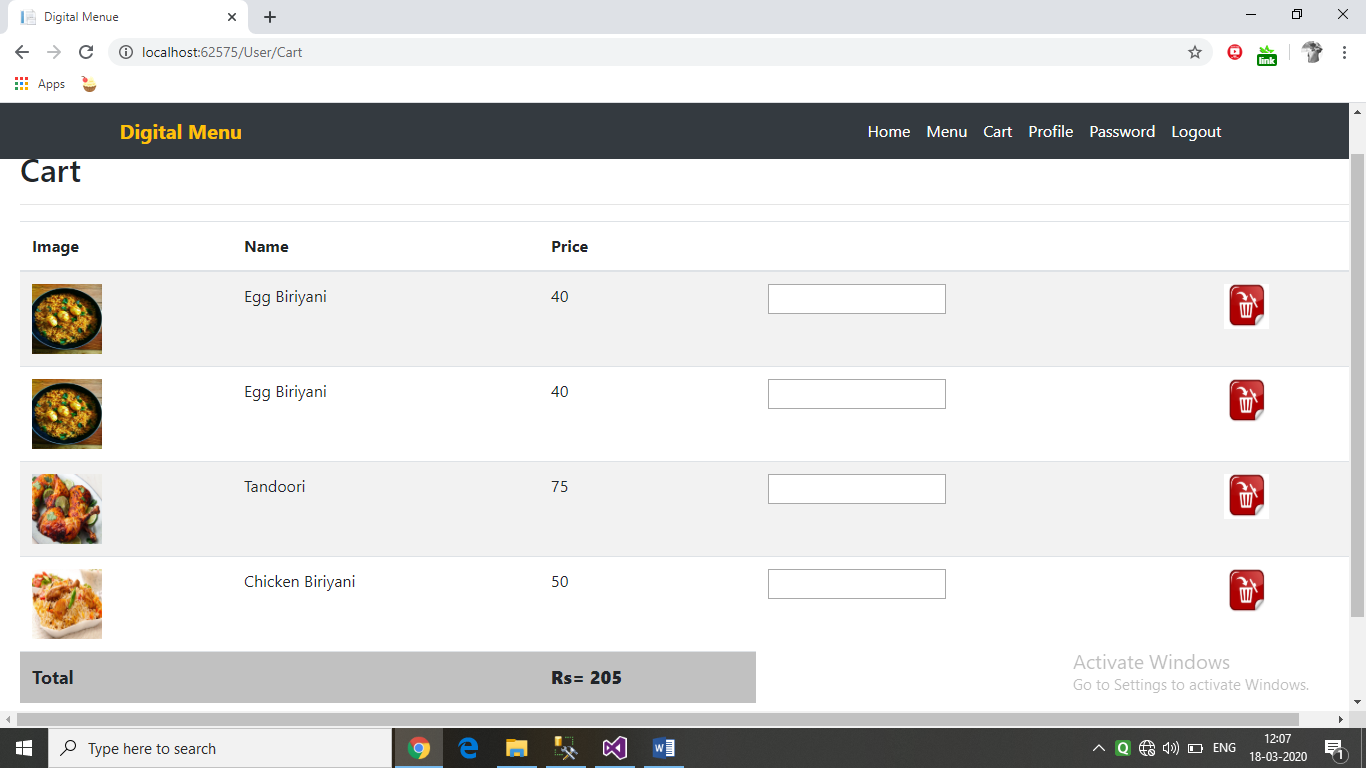


Figure 7.6: **Food Cart**

**CONCLUSION**

The proposed system of digital menu which is an online application is beneficial for the users that will help them to order food in the restaurants. Our application will also help the chef of the restaurant to order the required ingredients required for preparation of food by placing order to the farmers or retailers.

**FUTURE ENHANCEMENT**

In future, the application can be enhanced by taking this web based project to mobile application .So that the user can interface with more compatibility. This application can be implemented on AR which enables the user to change the apparent colour, texture, and even the size of the food they are eating .Simple multisensory dining experience by the display of type of videos which is accordingly to the dish you are eating

**REFERENCES**

**Referred Websites**

[www.wikipedia.com](http://www.wikipedia.com)

[www.codeproject.com](http://www.codeproject.com)

[www.csharp-station.com](http://www.csharp-station.com)

www.tutorialspoint.com

http://csharp.net-informations.com

[www.dotnetfunda.com](http://www.dotnetfunda.com)

[www.sqltutorials.com](http://www.sqltutorials.com)

www.sqltools.net

[www.uml-diagrams.org](http://www.uml-diagrams.org)

www.smartdraw.com

**Referred Books**

* “Begin ASP.NET 2.0 with visual C#.NET”, Wrox, By Chris Ullman.
* “Programming in C#, 3E” by BalagurusamyTata McGraw-Hill Education, 2010
* Software Engineering, Ian Summerville, Sixth Edition, Pearson Education Ltd, 2001
* Ian Sommerville, Software Engineering, Pearson Education, Sixth Edition
* ASP.net complete reference - Pearson Publication