

**KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES
TECHNOLOGY**

Fangyi Liu

Management of Hotel Dining Information System

The items that should remain confidential and the length of time of confidentiality are mentioned here.

The Bachelor's Thesis Information Technology programme
Kemi 2010

PREFACE

I want to thank my supervisor Aalto Teppo who gave me great help and good advice on my thesis modification.

I also want to thank my friend Bin Wang who assisted me in the project improvement.

ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology
Degree Programme Information Technology
Name Fangyi Liu
Title Management of Hotel Dining Information System
Type of Study Bachelor's Thesis
Date 1 September 2010
Pages 29 + 15 appendices or appendixes
Instructor Aalto Teppo
Company Kang Hua Company
Contact Person/Supervisor
from Company Kun Liu

Dining hotel management is more completed, more accurately, and effectively method for hotel and dining industries. It is an application based on modern hotel market. The information management system is already widely used in many fields at moment. Nowadays, with the dining hotel industries development, an advanced and scientific management system that could be more detailed and secure is needed. Hotel Dining Management system is extremely practical in the present hotel industry dining management. The system can do well in the hotel dining supervisory work management, the manager can also use computer to operate all kinds of data. The differences between computer and manual are that the computer is mechanical and precise. It has little probability to make mistake. Using computer to control dining hotel information has more advantages than by hand.

This work is done according to the software engineering principle, introduced this system design and the realization process, and made the specification about the demand analysis. It also presented the system outline design, designed in detail and the concrete realization. This system is basic on theory and method of management information system, supported by database technology. It might reduce staff's labour intensity, improve work efficiency and the management level, and make the enterprise to have better survival in the fierce market competition.

As the result of this work Administrator can update the food name, booking information and food material names; he also can add system operator information and backup the database. People use this system to manage the hotel's dining.

The main developing tools and languages are Visual Studio 2010, SQL Server 2008, ASP.NET and UML.

Key words: Hotel Dining Information System, ASP.NET, SQL Server 2008, ASP.NET

TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1.	Background	1
1.2.	Research problem	1
1.3.	Research goal	2
2.	TECHNOLOGY AND METHODS PREVIEW	3
2.1.	Visual studio 2010	3
2.2.	ASP.NET	4
2.3.	ADO.NET	4
2.4.	SQL Server 2008	5
2.5.	UML	6
2.6.	Software Requirements Specification	7
2.7.	Database Design	7
2.8.	GUI Design	8
3.	REQUIREMENTS ANALYSIS	9
3.1.	Requirements Determination	9
3.1.1.	Purpose and scope of project	9
3.1.2.	Project Perspective	9
3.1.3.	Stakeholders	9
3.1.4.	System services and constrains	9
3.2.	Functional Requirements	10
3.3.	Non-functional requirements	13
4.	DATABASE DESIGN	14
5.	GUI DESIGN	15
6.	SYSTEM IMPLEMENTATION	21
7.	CONCLUSIONS	27
8.	REFERENCES	28
9.	LIST OF APPENDICES	30

1. INTRODUCTION

1.1. Background

With the development of science and technology, the way of managing hotel dining is more and more advanced. A manual has many weaknesses, like management inefficiency, labor intensivity, the slow speed of manage information and accuracy rate is not satisfying. In order to improve the efficiency of hotel dining management, reduce labour intensity, raise information processing and accuracy, I choose the case of using computer to design hotel dining management system that could supply more convenience for user. The computer can automatically manage the hotel dining, the manager can use computer to implement hotel information management. The automation comes true. To be a successful hotel, the operator not only will improve service level and quality, it can raise market occupancy rate and have more returning customers; as well as have good work efficiency to control the costs. In information era, more important is must have one faultless management information system. In this study I design a hotel dining management system for a company and choose it to be my graduation project.

Through my studies, I already have several projects relevant to database systems. These courses, including Databases, Software Design Project and Programming Project, and the other basic courses, like Software engineering, Software Design Project and Software and System Testing, are pivotal experiences to acquire knowledge and achieve the goal of my thesis.

Based on my knowledge and interest, doing a hotel dining management system is an impressive and up to date topic. I can reinforce knowledge and acquire experience via this project and hotel can manage their dining information using the result of this project.

1.2. Research problem

“Compelling research problems can catalyze research process to be performed accurately and is cornerstone of any research methods and experimental design” (Martyn Shuttleworth, Dec 2008). It means that research problems can enable me to create a user-friendly and reasonable system. During the working process, I intend to answer the following research questions:

- What functionalities can be realized?
- How to design the system structure and layouts?
- What kind of data is needed to collect?

- How to establish the relationship of each module?
- How to implement the whole system?

1.3. Research goal

The goal of my project is to design and create a system, including user interface and databases, which users can manipulate and manage easily. This project hotel dining management system has three parts:

- Food management. Including two modules, food name information management and booking information management.
- Stock management. Including food modify module and raw material modify module.
- System management. It consists of Administrator information management module and back-up database module.

In order to achieve such a system, the procedure will be performed step by step presented in the following sidebars:

- Requirements analysis
- Database design
- User interface design
- System implementation and testing

The first phase concentrates on system's appearance, functions and client's requirements. I collect information and requirements from the customer and manager.

The second phase is to define the database, for example, which data has the relationship with the requirements, which data should be stored. I use entity relationship diagram (ERD) to draw the databases.

The third phase is that I make a multi-functional user interfaces so that users can operate easily. I use Visual Studio 2010 to create it.

Lastly, the implementation and testing phase is that make the users interfaces connect with database, test it whether can be execute successful.

2. TECHNOLOGY AND METHODS PREVIEW

2.1. Visual studio 2010

Microsoft Visual Studio is a complete set of development tools. It is used to build ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. All platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Basic, Visual C#, and Visual C++ all use the same integrated development environment (IDE). In addition, these languages use the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web Services. /1/

People can incorporate many technologies in the applications when they create by using Visual Studio. Such as .NET Framework, Windows Presentation Foundation (WPF), Windows Workflow Foundation, Silverlight, Windows Forms, ASP.NET, Extensible Application Markup Language (XAML) and so on. Visual Studio provides many application templates to help people create programs, and several programming languages in which to write them. Built-in languages include Visual C/C++, VB.NET, Visual C# and Visual F#. Support for other languages such as Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. The application templates consist of Windows applications, Web applications, Office applications, SharePoint applications and Extensibility applications.

Visual Studio 2010 has many improvements, like speed up code authoring, simplify Web deployment. In Visual Studio 2010, it includes Code Snippets for HTML, Jscript, and ASP.NET controls to help people write code faster. People can insert snippets from the Code Snippets Manager or directly from IntelliSense. Simplify Web deployment means user can package and publish their Web application in one click. The web deployment tool enables user to package their web application for deployment to an Internet Information Services (IIS) Web server. A Web package contains Web content, IIS settings, database scripts, components, registry settings and certificates. It can be a compressed file or a folder structure. The web deployment tool has been integrated into Visual Studio and enables user to create Web packages with one click. User also can now configure his project to transform the web.config file during deployment. When user deploy the project, the setting in web.config automatically match the settings on his debug, staging, and production servers. /2/

2.2. ASP.NET

ASP.NET is a web development model that could support user to build enterprise web applications with a minimum of coding. ASP.NET is a part of the .NET Framework, and user can access in the .NET Framework when coding ASP.NET applications. User also can code his applications in any language compatible with the common language runtime (CLR), such as Microsoft Visual Studio and C#. These languages could bring some benefits for the users developing ASP.NET application. /3/

People can use ASP.NET web pages to create dynamic content for their website. With a static HTML page what is .htm and html file, the server achieves a web request by reading the file and sending it to the browser. In contrast, when someone requests an ASP.NET web page that is .aspx file, the page runs as a program on the web server. The page can perform any task that user's website requires while the page is running. The task includes calculating values, reading or writing database information, or calling other program. As its output, the page dynamically produces markup and sends this dynamic output to the browser.

In ASP.NET Web pages, user interface programming is separated into two parts: the visual component and the logic. They mean the visible portion of a page and the code behind the page that interacts with it.

The logic for the ASP.NET Web page consists of code that you create to interact with the page. The code can reside either in a script block in the page or in a separate class. If the code is in a separate class file, this file is referred to as the code-behind file. The code in the code-behind file can be written in Visual Basic, Visual C#, Visual J#, and JScript .NET. /4/

ASP.NET Web pages are compiled into a dynamic-link library (.dll) file. The first time a user browses to the .aspx page, ASP.NET automatically generates a .NET class file that represents the page and then compiles it. The .dll file runs on the server and dynamically produces the HTML output for your page. /4/

2.3. ADO.NET

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed through OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data. /5/

ADO.NET uses two types of objects to access the data in a database: datasets, which can contain one or more data tables, and .NET data provider objects, which include data adapters, commands, and connections. /6/

There are four important components. They are Connection, Command, DataReader and DataAdapter. Connection means object supply the connection with data source. Command means developer can use it to return the data, modify data, run stored procedure, send and search the information. DataReader can provide high-powered data stream from the data source. At last, DataAdapter offers the connection of DataSet Object and data source. DataAdapter uses Command to execute SQL Order in data source, uploads the data to DataSet, and brings into correspondence with changed data from DataSet and data source.

2.4. SQL Server 2008

Microsoft SQL Server is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL. SQL Server 2008 aims to make data management self-tuning, self organizing, and self maintaining with the development of SQL Server. It also includes support for structured and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. SQL Server 2008 can be a data storage backend for different varieties of data: XML, email, time/calendar, file, document, spatial, etc as well as perform search, query, analysis, sharing, and synchronization across all data types. /7/

Words listed below used in SQL Server often help to familiarize the construction and features of SQL Server.

Relational Databases: “it is a set of tables used for storing data. Each table has a unique name and may relate to one or more other tables in the database through common values”. /8/

Table: a table is the basic units for data storage and organization in a database. One database contains many tables which are made up of fields containing records of data.

Relationship: it is a link between two tables; it can find data in one table that pertains to a specific record in another table.

Primary Key: it is highly recommended to create a primary key so as to identifying a unique record in a table and building connections for the tables.

SQL Server can build a database which is software for creating, manipulating and administering a database.

2.5. UML

“UML, or Unified Modeling Language, is a specification language that is used in the software engineering field. It can be defined as a general purpose language that uses a graphical designation which can create an abstract model. This abstract model can then be used in a system. This system is called the UML model”. /9/ It contains documentation that drives the model elements and diagrams, such as written use cases.

UML diagrams represent two different views of a system model:

- Static (or structural) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- Dynamic (or behavioral) view: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams. /10/

“UML is a visual language for specifying, constructing, and documenting the system. You can use UML with all processed, throughout the development lifecycle, and across different implementation technologies”. /11/

“The use case is a powerful and valuable tool to ensure a system’s usability and feasibility. It visualizes a system’s behavior like how it works from a user’s standpoint”. (Joseph, 2002) Hence, the use case diagram can describe the working process of our system based on the customer’s needs. A use case diagram is not just a diagram but also a fully documented model of the system’s intended behavior.

Activity diagrams are a means of describing workflows and can be used in a variety of ways. As an analysis tool they can describe business flows in varying levels of detail. They can also be used to describe complex flows within or between use cases. At the design level, they can be used to describe in detail the flow within an operation. In this sense, they are very flexible. They can be used before the identification of use cases in the determination of high level business requirements, as a means of describing complex use cases and as a means of describing complex behaviour within an object. /12/

The activity diagram shows the steps of a computation. These steps representing a behavior from an actor are called actions. The flow of control from one action to the next is called a control flow. The activity diagram begins with a starting point the symbol of which is a filled-in circle and finishes with an endpoint represented by a bull’s eye. /13/

2.6. Software Requirements Specification

A Software Requirements Specification (SRS) is a description of the developed behavior system. It includes functional requirements and non-functional requirements. Functional requirements describe all the interactions the users will have with the software. "Non-functional requirements are requirements which impose constraints on the design or implementation, such as performance engineering requirements, quality standards, or design constraints". /14/

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. /15/

The non-functional requirements define many aspects about the system. These characteristics include how easy the software is to use, how quickly it executes, how reliable it is, and how well it behaves when unexpected conditions arise. /16/

The non-functional requirements should be defined precisely. It should provide specific measurements that the software must meet. The maximum number of seconds it must take to perform a task, the maximum size of a database on disk, the number of hours per day a system must be available, and the number of concurrent users supported are examples of requirements that the software must implement but do not change its behavior./16/

2.7. Database Design

The database is an indispensable part of life that we are using all the time like checking a bank account and so on. A database is an organized and shared repository of data which includes data itself and data descriptions stored in the computer. Data of a database is managed by using a certain data model so that the database is accompanied with smaller redundancy, high degree of data independency and flexible expansion. DBMS stands for database management system which is software to administrate and control access to the database. A database application is a program that interacts with a database. /17/

As a database is a fundamental component of a database management system it is vital to recognize the database applications used for realizing the system and the stages of the database application lifecycle for designing the system. Three essential phases of database design are as follows:

1. Conceptual database design to use the Entity-Relationship diagram to build a conceptual data model including identification of the important entities, relationships and attributes

2. Logical database design to interpret the conceptual model to the logic structure of the database which includes the design of the relations
3. Physical database design to produce a description of the implementation of the database /17/

2.8. GUI Design

A graphical user interface, the abbreviation of which is GUI, is a platform based on icons, pictures and menus instead of text and uses a mouse as well as a keyboard to realize the interaction between human and computer. It is a human-computer graphic system with combination of computer science, aesthetics, psychology, behavioral science and business analysis. Objects and actions are used to comprise this graphical system. Objects manipulated as a single unit are what people see on the screen. A well-designed system keeps users focused on the objects instead of on how to carry out actions. /18/

GUI design is a multidisciplinary activity requiring a diverse mix of skills ranging from visual art to software programming. Depending on heuristics, which are the general principles of thumb for user interface designs the core essential issue of designing GUI is that user is in control, however consistency is the key to solve user's control and interaction. /19/

Benefits of Good Design are presented following below:

- Higher task completion rates
- More efficient task completion rates
- Reduced training costs
- Improved customer service /18/

3. REQUIREMENTS ANALYSIS

3.1. Requirements Determination

3.1.1. Purpose and scope of project

The purpose of this project is to create a system to manage the hotel dining ordering and stocking information more convenient.

3.1.2. Project Perspective

Hotel Dining Information Management System (HDIMS) should be able to provide a basic and easy way to manage the information. When user apply this system, will know hotel details, like which food should be stocked. Remove the communication gaps between customers and hotel staff. It should be compatible with all the operating systems.

3.1.3. Stakeholders

This system is accessed by limited people with limited rights. Users of this system are administrators, waiters and hotel staff. All the users can log in system. Waiters are in charge of food management. Staff is responsible for stock management. Administrators are responsible for system management.

3.1.4. System services and constrains

The system includes food name information, ordering food information, administrator information and back-up database. The vital functions should be realized in this system:

- Add
- Delete
- Modify
- View

3.2. Functional Requirements

The following figure1 illustrates behaviors of the system and relationships among actors and use cases and generalization association between actors. As an administrator, not only he or she view and modify the data that he or she wants, but also manage all the data of database. Waiter manages food data, and staff manages stock data. Apparently the administrator is the main person to operate all functions of this system.

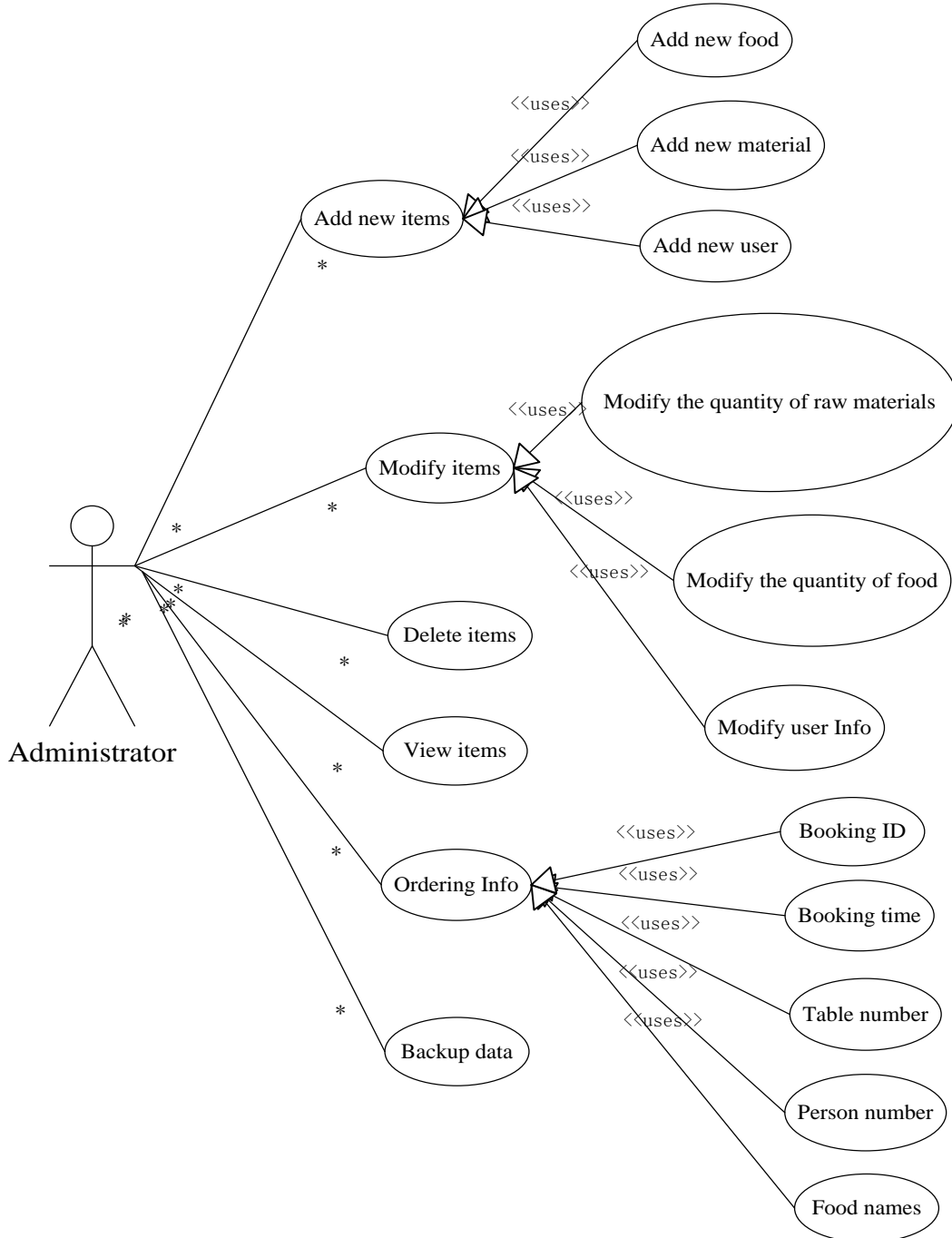


Figure 1. User Function Structure

Add new items means users record the data about adding food, material and new user information into the database system. If the table that is requested to fill is empty, the system asks if it rewrites or close the window; if the format in the table is not right, the system asks if it rewrites or close the window.

Modify items means database system authorizes the right of users and allows he or she to modify data. If user types the data into a wrong table, the system will show an error message; if user type the data in a wrong format, the system will show an error message.

Delete items is users delete useless data. Users find the items that need to be deleted, then click the button "delete", the system asks if user really wants to delete the data of this item, users click "yes" and the data of this item is deleted permanently.

View items is database system which authorizes the ID of a user and allows the user to view data. If user type wrong username or password, the system shows error message and requires a user to re-log in.

Ordering items means that user records the data of booking food into the system. If user types the data into a wrong table, the system will show an error message and asks if it overwrites; if user types the data in a wrong format, the system will show an error message and asks if it overwrites.

Backup data means administrator to keep a backup of system database. Before operating backup data, the system verifies the right of an administrator, if users have right, the proper data will be shown, and administrator clicks the button "backup data", the data stored into the system quickly.

The actions of user are elaborated in activity diagram drawn in Figure2. It demonstrates that user's rights to operate the system. Apart from the actions of "viewing" "booking" and "backup data", the user can modify the data of the system like modifying, adding and deleting. The database system saves all changes made by the user. In the activity diagram, a synchronization bar is used to split user actions to detailed single activities and then these activities are assembled together to another bar. The relation of these activities is parallel instead of being sequent. It means that a user can do either one of the actions or all of actions. The last step is to log out the system after all these actions.

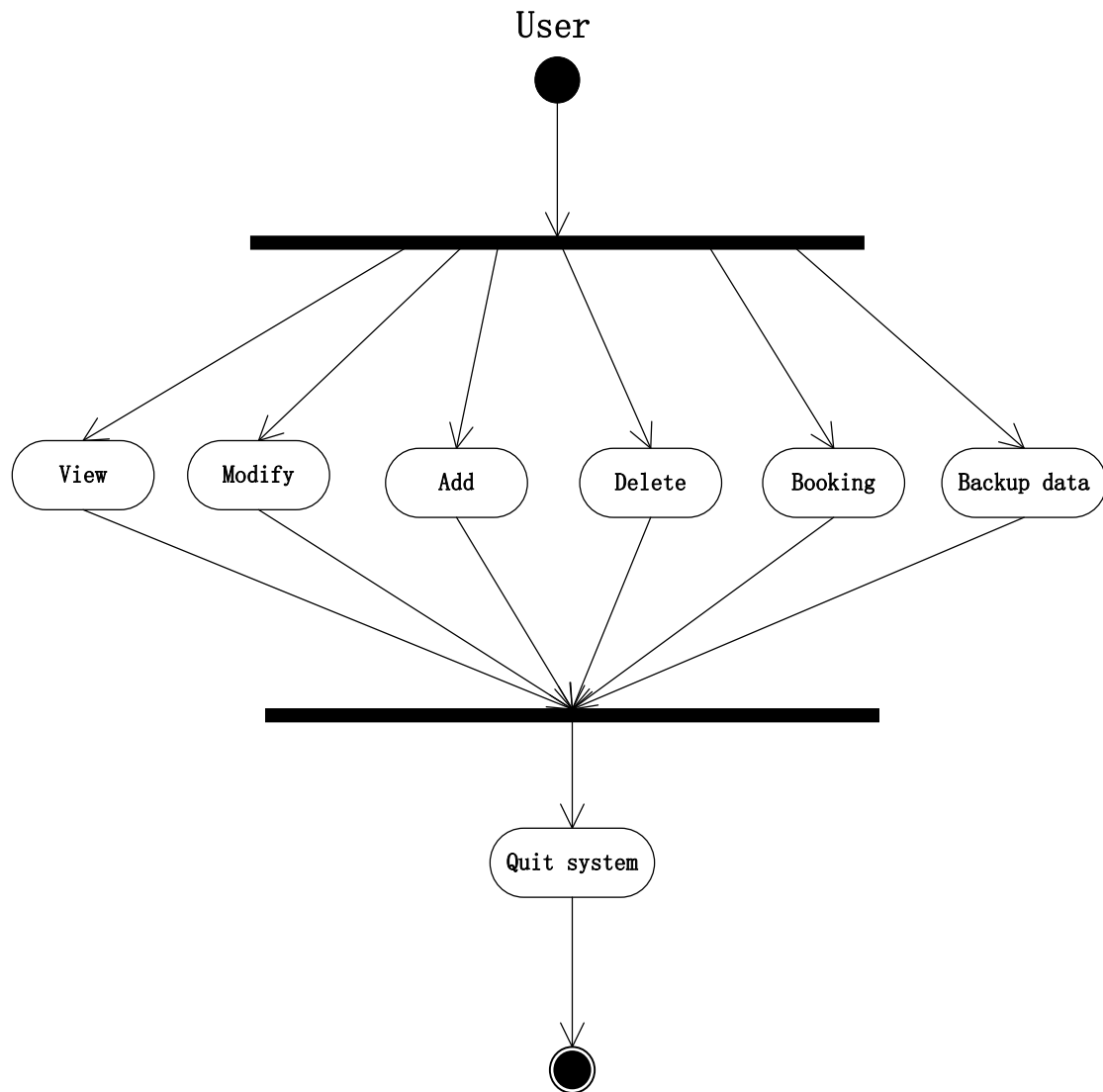


Figure 2. System Structure for use

3.3. Non-functional requirements

The non-functional requirements of this system include some important items. Reliability, Maintainability and flexibility have close relationship. This system can be recovered automatically when some problems appear or users have incorrectly manipulated. It also can point out the wrong position when users encounter some problems in using system. Such as users write wrong format information in system, it will be noticed and tell users where the wrong information is. This system has some appropriate measures to protect data when close the application unusually too. Like when users open the system after it closed unusually one time, it will ask users whether they want to continue last operation or not. So the Reliability, Maintainability and Flexibility can solve these questions above mentioned.

Usability is another element for this system. The functions of the system can be easy to use for the users, and then users could get an expected result after running a specific function. Such as the user interface has clear navigation, which could direct users how to use the system. No matter how much computer knowledge users know, they can operate it very well.

If it considers both reliability and maintainability, item availability is needed too. Availability of this system means the running time of system can be used 99.5%. The response time is less than 5 seconds.

Safety includes preventing from illicit access system, data lose, and virus invades. It will protect the private information of users and hotel. Nobody can check private thing except system users. The compatibility of system is it can be used in any computer system and compatible with any software. Resource constraints are some hardware requirements, for example, CPU is at least Inter Pentium4 Dual Core Processor, the computer memory is at least 512M, and computer hard disk is at least 40G. Privacy is specific user uses their username and password to check specific information.

4. DATABASE DESIGN

The ER diagram is used to model the conceptual database for database system. It is easy and convenient way to visualize a complex system instead of narrative texts. The diagram indicates the structure of the database as below.

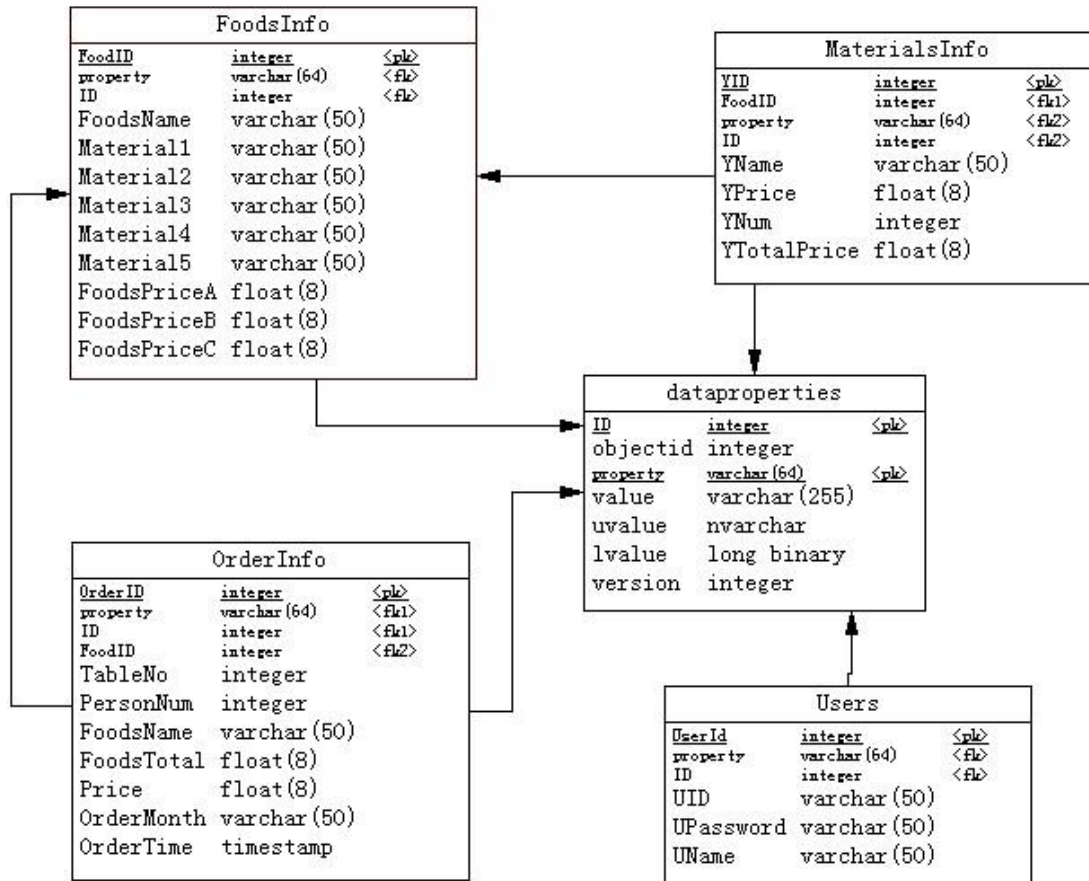


Figure 3. Entity-Relationship diagram

FoodsInfo is a table to save detailed information concerning foods name, material and price.

MaterialInfo table includes material names, price, number and total price, which is connected to the tables of FoodsInfo and dataproperties.

OrderInfo, which stores order data such as table number, person number, food name, total foods, total price and order time, which is connected with FoodsInfo table.

Users table is to save user information, like user id, password, and name.

5. GUI DESIGN

The database management system will be used to manage information of the commissioner, so users with limited computer knowledge should easily manipulate user interfaces. In this case the user interfaces were designed with fully fundamental functions with concise words instead of being aesthetic. Figure4 shows the first step login while using the system.

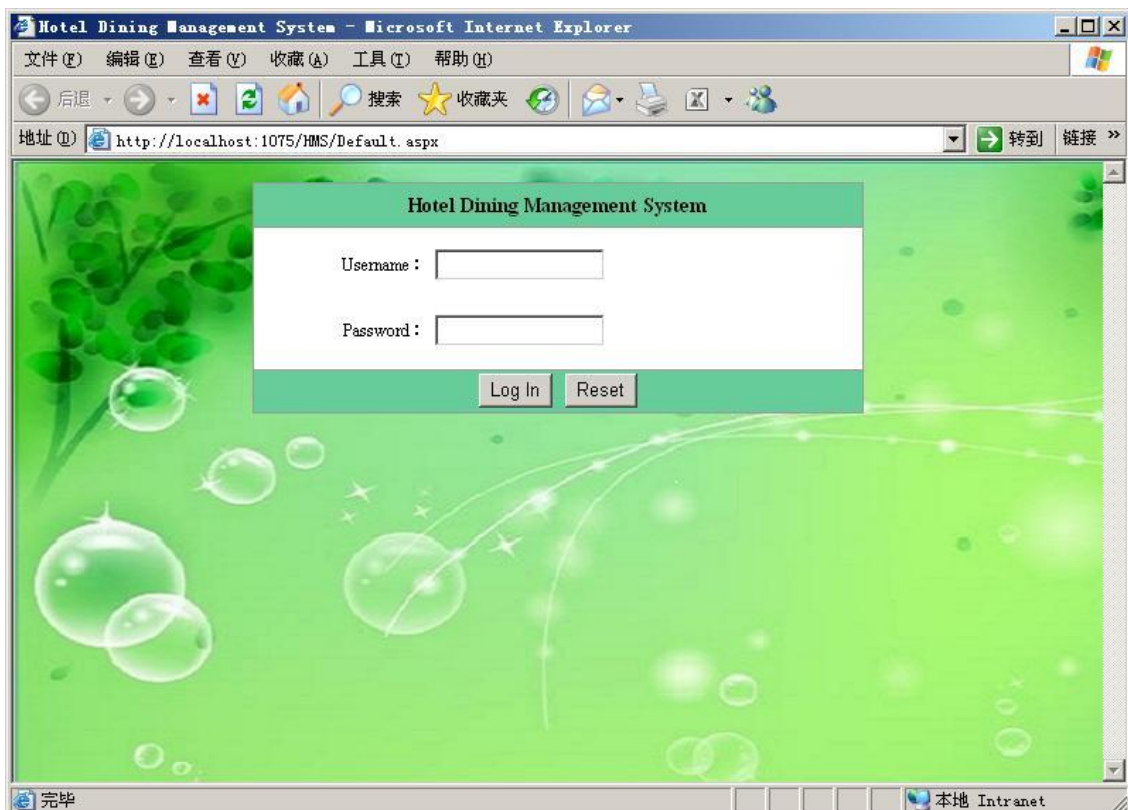


Figure 4. Login user interface

A main user interface will appear after logging into the system with six different categories. As Figure 5 shows as below all buttons of these five categories are placed on the left side from up to down. A user can enter the sub-interfaces by clicking one of these buttons.

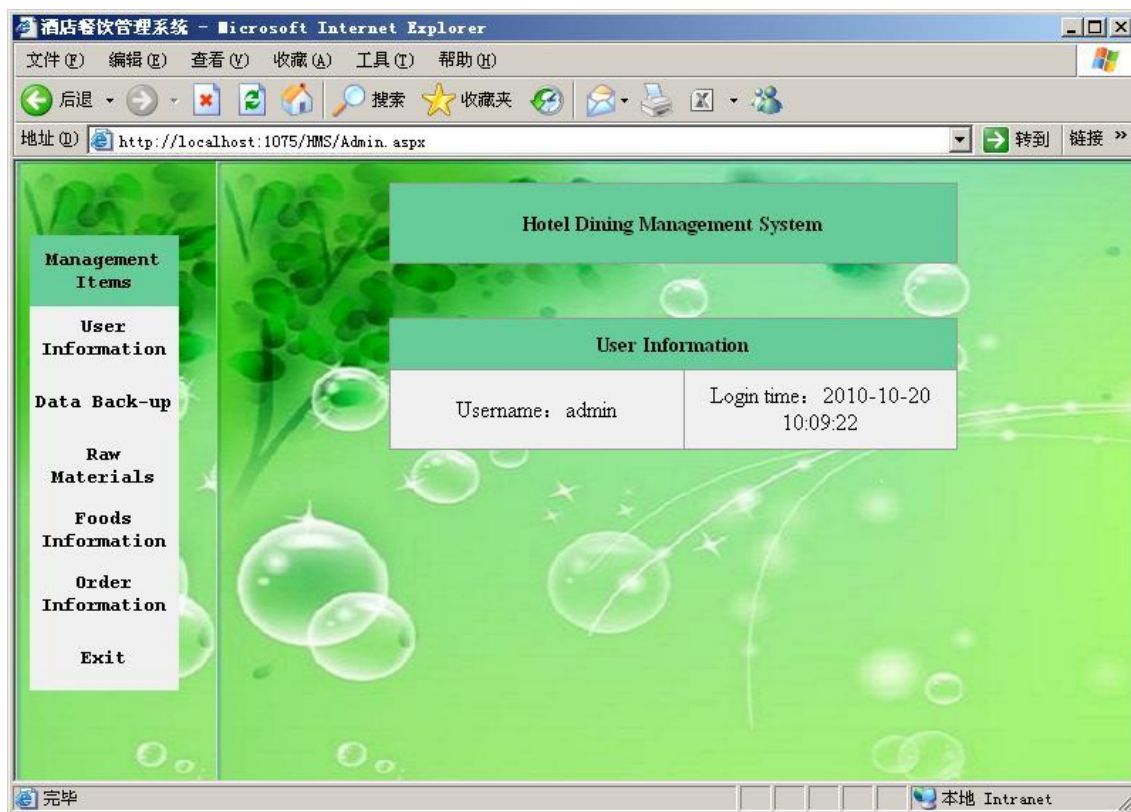


Figure 5. Main user interface

Figure6 shows the user information list interface. User can add new user and search user information. In the user Info list, people can check User ID, user name, password, detail Info and delete user.

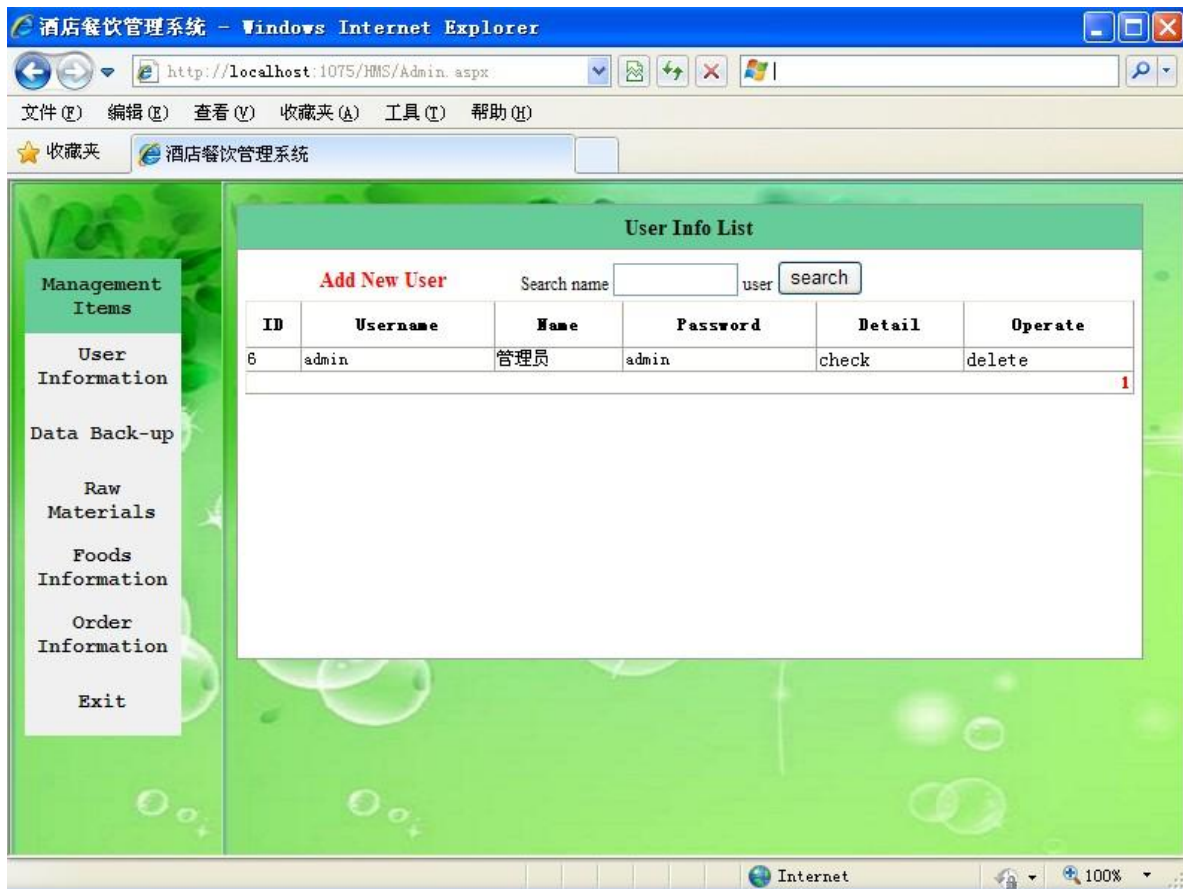


Figure 6. User Info interface

Figure7 shows the Raw Materials user interface. There is material information, such as materials ID, name, details and operate. If user wants to add new material, he can click "Add new food" button. Then it will jump to another add material Info interface, user can put material name, price and number in the table. User can do delete if he do not need that material too.



Figure 7. Raw Materials user interface

Figure8 shows the information about food ID and name. If user does not need food information, he can press “Delete” in the table.

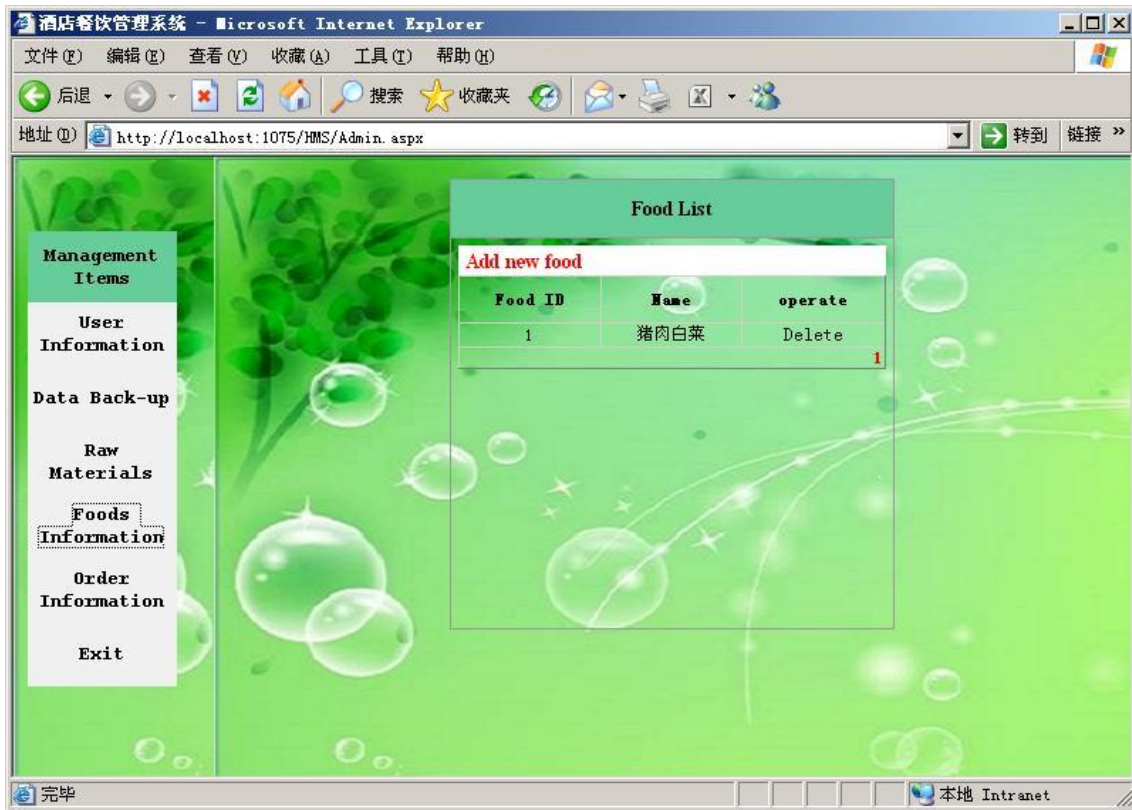


Figure 8. Foods Information user interface

As Figure9 shows below the information is order foods. There are order ID, table number, food name, total price and booking time. If user wants to add new order, he can click red button “Add new order”.

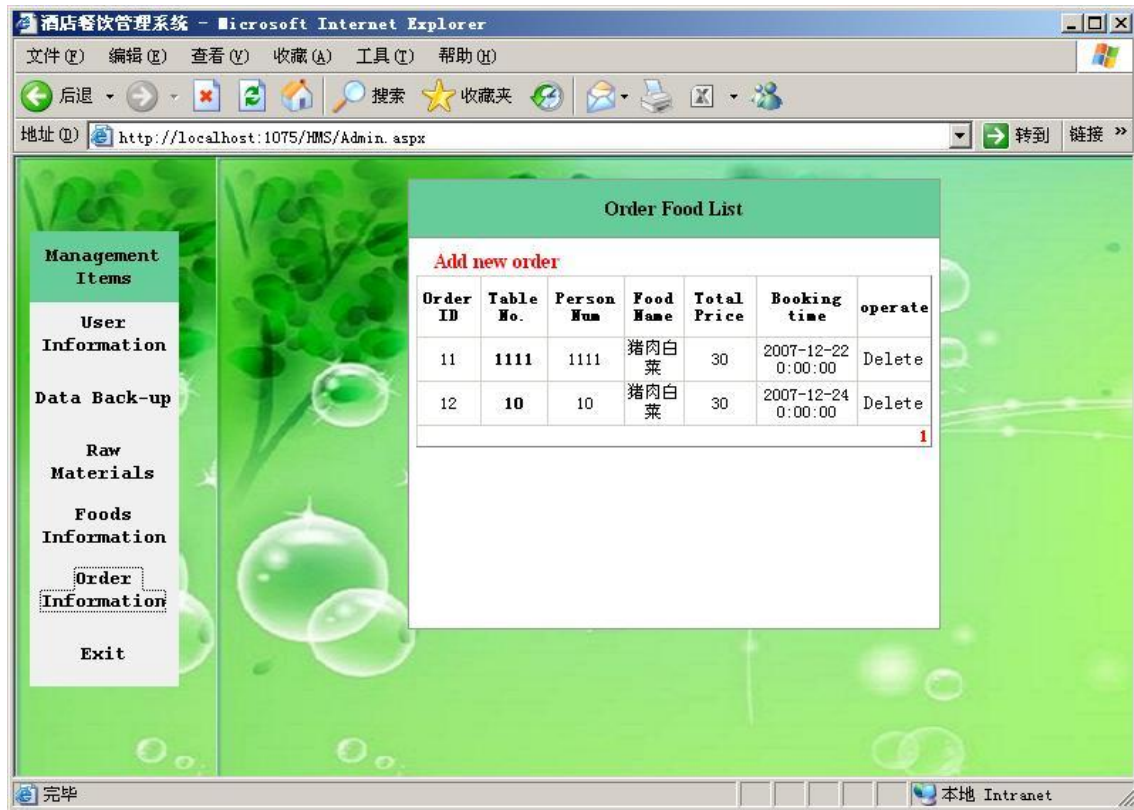


Figure 9. Order Information user interface

6. SYSTEM IMPLEMENTATION

Another pivotal step of this project is to implement appropriately the database management system. This section will introduce execution of the fundamental functions of the system and technologies chosen for the system.

After logging in the system the user jumps to the main interface. There are six main buttons including Management Items, User Information, Data Backup, Raw Materials, Foods Information and Order Information.

The function of adding new user is used here to be example. The User Information List interface is mentioned in GUI design before. There is one button "Add New User", clicking it then a sub-interface appears. The user can save new user information in the sub-interface presented in Figure 10.



The screenshot shows a web form titled "Add New User" with a green header. The form contains four text input fields labeled "user name", "password", "confirm password", and "name". At the bottom, there is a green bar with a "submit" button and a "back" link.

Figure 10. Sub-interface for adding new user

Below is the code for checking whether username is registered or not. `Args.IsValid = False` means the username is registered, otherwise it is not registered.

```
if (dr.Read())
{
    args.IsValid = false;
}
else
{
    args.IsValid = true;
}
```

In Material List interface there is one button “Add new material”, clicking it then a sub-interface appears. User can input new material Info and save them. Figure 11 shows the sub-interface for adding material.

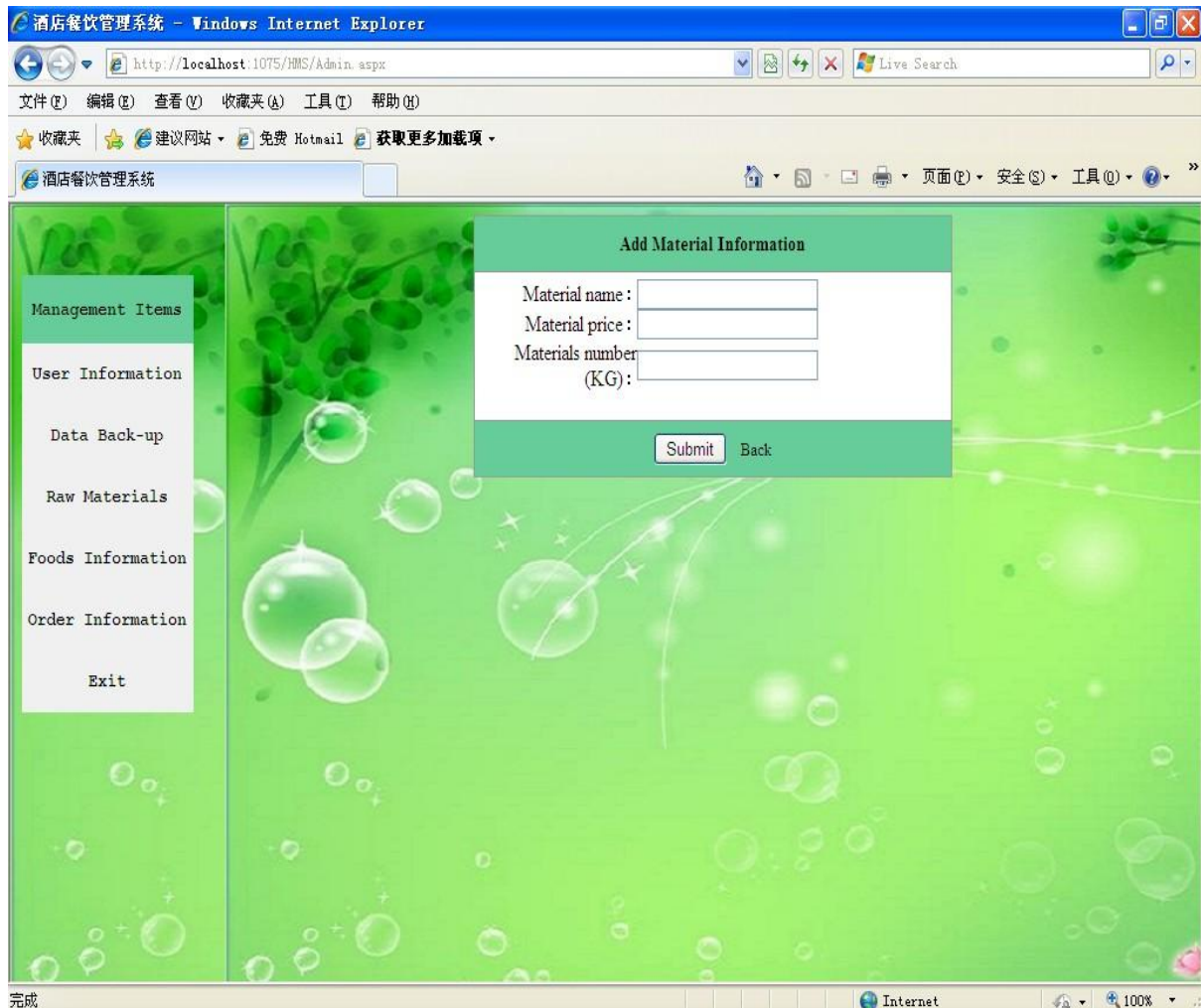


Figure 11. Sub-interface for adding material

The code is showed below means adding some parameters into stored procedures. VarChar, Float and Int are the class of material name, price and number.

```
cm.Parameters.Add("@YName", SqlDbType.VarChar);  
cm.Parameters.Add("@YPrice", SqlDbType.Float);  
cm.Parameters.Add("@YNum", SqlDbType.Int);  
cm.Parameters.Add("@YTotalPrice", SqlDbType.Float);
```

Assign values to variables in stored procedures:

```

cm.Parameters["@YName"].Value = tbx_tname.Text.ToString();
cm.Parameters["@YPrice"].Value = Convert.ToDouble(tbx_uname.Text);
cm.Parameters["@YTotalPrice"].Value = Convert.ToDouble(tbx_uname.Text) * Convert.ToInt32(tbx_num.Text);
cm.Parameters["@YNum"].Value = Convert.ToInt32(tbx_num.Text);

```

In Foods Information interface, user can click one button "Add new food" and save new food information in the sub-interface what will see after click button. Figure 12 is sub-interface for adding food. In this picture, food materials include five tables; user can choose material from the table. There are three materials such as Chinese cabbage, pork and egg were written as example in the database. The weight of material can be 0-10 kg; user can choose number in the table.

Food Name :			
Food Materials :	白菜	0	kg
	白菜	0	kg
	白菜	0	kg
	猪肉	0	kg
	鸡蛋	0	kg
	白菜	0	kg
Food Sale Price(RMB) :			
<input type="button" value="Submit"/> <input type="button" value="Back"/>			

Figure 12. Sub-interface for adding food

Below is the code for adding food interface connects localhost database and use Command Object to debug stored procedures. In this code, the sentence `SqlConnection cn = new SqlConnection ("Data Source=.\SQLExpress; Initial Catalog = HMS; Integrated Security = True")` is used to connect local database. HMS is the local database name. The other sentence `SqlConnection cn = new SqlConnection ("server=local; uid = kanghua; pwd = kanghua1988@; database=HMS")` was used before when compiling the connection, it is more complex than what is writing now. In

order to avoid more errors during working, the easier way is chosen to show. Because the edition of SQL Server is Express, code Data Source=.\SQLExpress is needed to write when introduce the data source.

```
string strconn = ConfigurationSettings.AppSettings["dsn"];
SqlConnection cn = new SqlConnection("Data Source=.\SQLExpress;Initial Catalog=HMS;Integrated Security=True");
cn.Open();
SqlCommand cm = new SqlCommand("select * from YuanLiaoInfo", cn);
SqlDataReader dr = cn.ExecuteReader();
while (dr.Read())
{
    ddl1.Items.Add(new ListItem(dr["YName"].ToString(), dr["YName"].ToString()));
}
cn.Close();
```

In Order Food List interface, there is a sub-interface for adding new order. Figure 13 exhibits the appearance of the user interface with the attributes. There are some codes for the adding interface will be listed as example, which is interesting. They are tightly flowing with Figure 13.

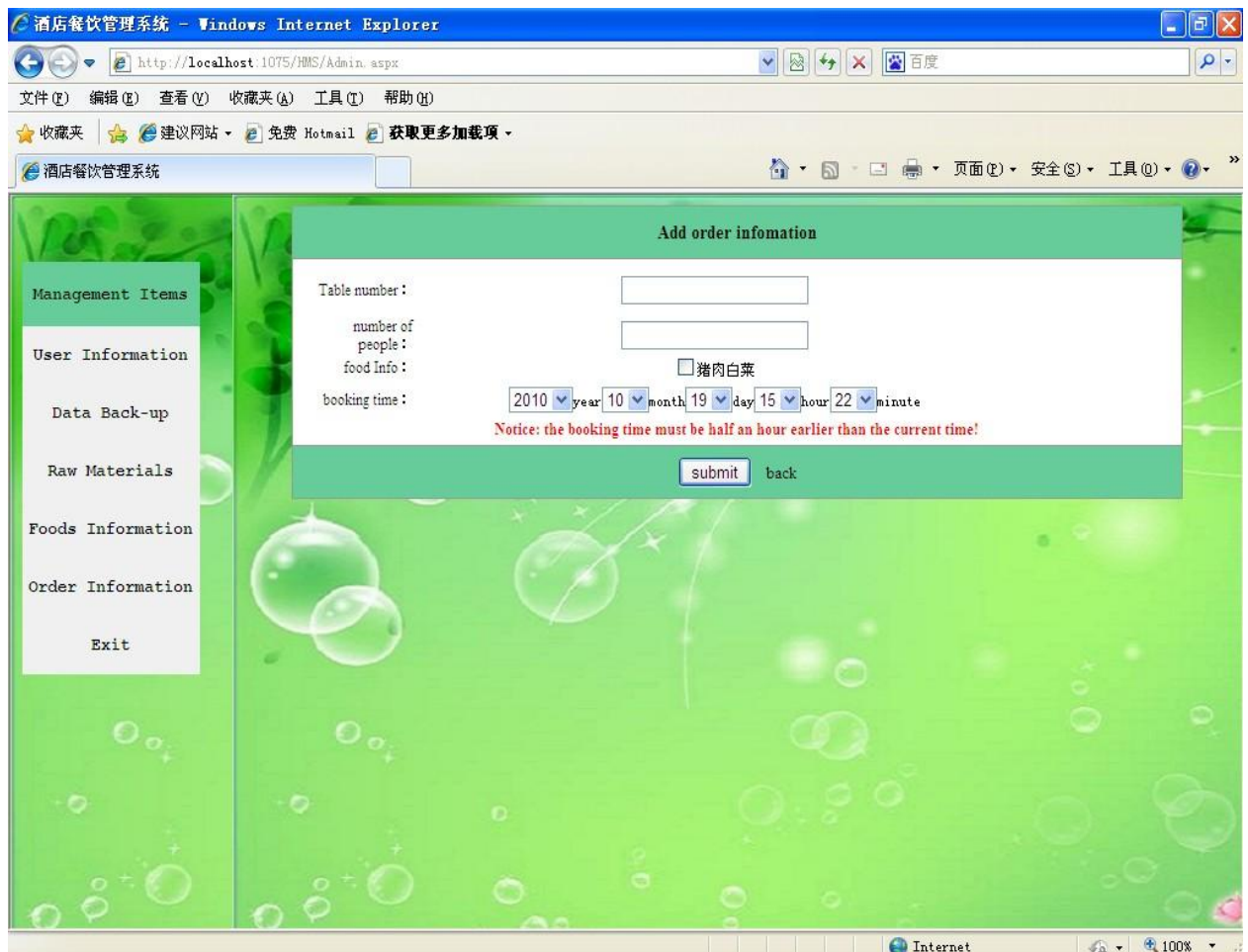


Figure 13. Sub-interface for adding order

The code below is adding and defining items like year, month, day, hour and minute. The scope time of year is from 2007 to 2020, month is from 1 to 12, day is from 1 to 31, hour is from 0 to 23, and minute is from 0 to 59. They are gradually increasing. User can choose any one of them from the tables.

```
for (int i = 2007; i <= 2020; i++)
{
    ddl_Y.Items.Add(new ListItem(i.ToString(), i.ToString()));
}

for (int i = 1; i <= 12; i++)
{
    ddl_M.Items.Add(new ListItem(i.ToString(), i.ToString()));
}

for (int i = 1; i <= 31; i++)
{
    ddl_D.Items.Add(new ListItem(i.ToString(), i.ToString()));
}

for (int i = 0; i <= 23; i++)
{
    ddl_H.Items.Add(new ListItem(i.ToString(), i.ToString()));
}

for (int i = 0; i <= 59; i++)
{
    ddl_S.Items.Add(new ListItem(i.ToString(), i.ToString()));
}
```

The code showed below sets the format of date.

```
if (chkbl.SelectedValue == "")
{
    Response.Write("<script language=javascript>alert('Warning: Please choose the reserved food name! ');history.back();</script>");
    Response.End();
}

if (DateTime.Now.AddSeconds(30) > Convert.ToDateTime(ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value
+ "-" + ddl_D.SelectedItem.Value + " " + ddl_H.SelectedItem.Value + ":" + ddl_S.SelectedItem.Value + ":00"))
{
    Response.Write("<script language=javascript>alert('Warning: Please choose the right time! ');history.back();</script>");
    Response.End();
}
```

The code below is in order to create SQL execute statement; it checks whether username is existed or not in User list.

```
string mysql = "select * from OrderInfo where TableNO=" + Convert.ToInt32(tbx_TNum.Text) + " and OrderTime=" + Convert.  
    ToDateTime(ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value + "-" + ddl_D.SelectedItem.Value + " " +  
    ddl_H.SelectedItem.Value + ":" + ddl_S.SelectedItem.Value + ":00") + "'";
```

7. CONCLUSIONS

A bachelor thesis, which is done by student under a supervisor's guidance, is the landmark achievement of my study and also the method to evaluate the knowledge acquired from my courses and capacity to cope with the future work. To build a dining management system for hotel is the topic of the thesis to assess my study and ability, and furthermore the system is realized to be applied for the hotel. This task is a time-consuming job because of its complexity. Before doing it, I spent time in collecting feedback from customer who uses this system. Then design and modify the system. It cost me a lot of time. The know-how for programming is a necessary ability during the implementation of the system. I had to enrich the knowledge that I have known and learn what I did not know based on my current knowledge. I used Visual Studio 2010, SQL 2008, Visual C #language and Transact-SQL language. At last was testing the system. For the implementation, I solved many problems during the testing period. Some problems were the setting of connection with interface and database, and some problems were the configuration of the interface.

I succeeded in completing this project at last, although it was the first time for me to implement a project independently. During the process, I practiced my capability of independent study, and accumulated many successful and unsuccessful experiences which are the big riches to me via this project work.

There are some advantages of this system. Firstly, integrality. It includes many functions such as ordering food, raw material information, and data management. It meets the demand of current hotel. Secondly, practicability. It is convenient to add, modify, delete, and consult the information in the system. Thirdly, maintainability. System can backup and resume data anytime; it improves the feasibility of database maintaining.

Because of my insufficient knowledge and time my system might be vulnerable to security threats and functionally constrained. It is better to reinforce the security to prolong the validation of the system like updating SQL. This system will be improved; it can add more function such as human resource, finance and so on. The architecture of system is not very good, it wastes much system configure.

8. REFERENCES

List of the sources:

- /1/ Microsoft corp., [WWW-document],
<<http://msdn.microsoft.com/en-us/library/fx6bk1f4.aspx>>, 06.09.2010
- /2/ Microsoft corp., [WWW-document],
<<http://msdn.microsoft.com/en-us/library/dd547188.aspx>>, 06.09.2010
- /3/ Microsoft corp., [WWW-document],
<<http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>>, 06.09.2010
- /4/ Microsoft corp., [WWW-document],
<[http://msdn.microsoft.com/en-us/library/428509ah\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/428509ah(v=VS.90).aspx)>, 06.09.2010
- /5/ Microsoft corp., [WWW-document],
<[http://msdn.microsoft.com/en-us/library/h43ks021\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/h43ks021(VS.71).aspx)>, 10.09.2010
- /6/ Anne, Prince; Doug, Lowe, VB.NET Database Programming with ADO.NET, Murach Publishing, 2003
- /7/ The Code Online, [WWW-document],
<<http://www.gigyonline.com/code/source/sql.html>>, 10.09.2008
- /8/ webucator, SQL Tutorial, [WWW-document],
<<http://www.learn-sql-tutorial.com/DatabaseBasics.cfm>> 10.09.2010
- /9/ Jon Holt Institution of Electrical Engineers, UML for Systems Engineering: Watching the Wheels IET, 2004, ISBN 0863413544, p.58
- /10/ Object Management Group.Inc, [WWW-document],
<http://www.omg.org/gettingstarted/what_is_uml.htm> 17.09.2010
- /11/ IBM Software, [WWW-document],
<<http://www-01.ibm.com/software/rational/uml/>> 17.09.2010
- /12/ Simon, Bennett; John, Skelton; Ken, Lunn, Schaum's Outline of UML, McGraw-Hill Companies, 2001, p.208
- /13/ Leszek, A.Maciaszek, Requirements Analysis and System Design, 3rd edition, Addison-Wesley Professional, 2007
- /14/ CECSOft, SRS, [WWW-document],
<<http://www.cecssoft.com/thinking/documents.html#srs>> 18.09.2010

- /15/ Resources for Software Architects, [WWW-document],
<http://www.bredemeyer.com/pdf_files/functreq.pdf> 18.09.2010
- /16/ Andrew, Stellman; Jennifer, Greene, Applied Software Project Management, O'Reilly Media Publish, 11.2005, p113
- /17/ Thomas, Connolly; Carolyn, Begg, Database Solutions, 2nd edition, ADDISON-WESLEY, 2002
- /18/ Graphical User Interface Design, Fall 2006 (PPT) 18.09.2010
- /19/ Leszek, A.Maciaszek, Requirements Analysis and System Design, 3rd edition, Addison-Wesley Professional, p380-p382, 2007

9. LIST OF APPENDICES

Appendix 1: Specification of the use case diagram

Appendix 2: Code for web configure

Appendix 3: Code for connection with Login and database

Appendix 4: Code for connection with food management and database

Appendix 5: Code for database backup

Appendix 6: Code for adding new material

Appendix 7: Code for adding new ordering

APPENDIX 1/1:

Use Case ID:	1
Use Case Name:	Add food, material and new user Info
Created By:	Fangyi Liu
Date Created:	10 September 2010
Actors:	Administrator
Description:	Users record the data about adding food, material and new user Info into the database system.
Preconditions:	System interface and database are shown and users are allowed to add some items.
Normal Flow:	<ol style="list-style-type: none"> 1. Adding interface is shown 2. Filling new item in the table, like name, number 3. Click button "Save" 4. Information is saved in database
Alternative Flow:	<ol style="list-style-type: none"> 1. If the table is empty, the system asks if it rewrites or close the window. 2. If the format in the table is not right, the system asks if it rewrites or closes the window.
Post conditions:	The data of a new item is saved and the system returned back the primary shape and allows an administrator to continue to operate the system.

Table 1.1. Use case for add item

APPENDIX 1/2:

Use Case ID:	2
Use Case Name:	Delete food, material and user Info
Created By:	Fangyi Liu
Date Created:	11 September, 2010

Actors:	Administrator
Description:	Users delete useless data.
Preconditions:	The system shows the interface and database.
Normal Flow:	<ol style="list-style-type: none"> 1. Find the items need to be deleted 2. Click the button “delete” 3. The system asks if user really wants to delete the data of this item 4. Click “yes” and the data of this item is deleted permanently
Alternative Flows:	None
Post conditions:	The system stores the change.

Table 1.2. Use case for delete item

APPENDIX 1/3:

Use Case ID:	3
Use Case Name:	Modify existed data
Created By:	Fangyi Liu
Date Created:	11 September, 2010

Actors:	Administrator
Description:	Database system authorizes the right of users and allows him or her to modify data
Preconditions:	User logs in and the system shows the interface and database.
Normal Flow:	<ol style="list-style-type: none"> 1. The system verifies the right of user 2. A user interface and database are shown 3. User types the data which needs to record into the database 4. The data is stored into the system successfully
Alternative Flows:	<ol style="list-style-type: none"> 1. Type the data into a wrong table, the system will show an error message 2. Type the data in a wrong format, the system will show an error message
Post conditions:	After storing the data, the system returns back the primary shape and allows a user to continue to operate the system.

Table 1.3. Use case for modify existed data

APPENDIX 1/4:

Use Case ID:	4
Use Case Name:	View information
Created By:	Fangyi Liu
Date Created:	11 September, 2010

Actors:	Administrator
Description:	Database system authorizes the ID of a user and allows the user to view data.
Preconditions:	A user logs in the system with proper personal ID and the system shows appropriate functions to the user.
Normal Flow:	<ol style="list-style-type: none"> 1. Type username and password 2. Show the interface and database 3. Find the data which a user wants to check
Alternative Flows:	<ol style="list-style-type: none"> 1. Type wrong username or password 2. The system shows error message and requires a user to re-log in
Post conditions:	Proper data is shown

Table 1.4. Use case for view item

APPENDIX 1/5:

Use Case ID:	5
Use Case Name:	Ordering food information
Created By:	Fangyi Liu
Date Created	11 September, 2010

Actors:	Administrator
Description:	User records the data of booking food into the system
Preconditions:	User interface and database are shown
Normal Flow:	<ol style="list-style-type: none"> 1. Fill the detailed information in the table 2. Click the button "Save" 3. The database saves the data of the table
Alternative Flows:	<ol style="list-style-type: none"> 1. Type the data into a wrong table, the system will show an error message and asks if it overwrites 2. Type the data in a wrong format, the system will show an error message and asks if it overwrites
Post conditions:	The data is saved, the system returns back the primary shape and allows a waiter to continue to operate the system

Table 1.5. Use case for booking food information

APPENDIX 1/6:

Use Case ID:	6
Use Case Name:	Backup data
Created By:	Fangyi Liu
Date Created:	11 September, 2010

Actors:	Administrator
Description:	Administrator keep a backup of system database
Preconditions:	All the data is stored in system database successfully and completely.
Normal Flow:	<ol style="list-style-type: none"> 1. The system verifies the right of an administrator 2. The proper data is shown 3. Administrator clicks the button "Backup data" 4. The data is stored into the system
Alternative Flows:	None
Post conditions:	After storing the data, the system returns back the primary shape.

Table 1.6. Use case for backup data

APPENDIX 2:

```

<configuration>
  <connectionStrings>
    <add name="HMS_DataConnectionString" connectionString="Data Source=.\SQLExpress;
      AttachDbFilename=C:\Program Files\Microsoft SQL Server\MSSQL10.SQLEXPRESS\MSSQL\DATA\HMS.MDF;Integrated Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="dsn" value="Data Source=localhost;UID=kanghua;PWD=Kanghua1988@;DATABASE=HMS"/>
  </appSettings>
  <system.web>
    <!-- 动态调试编译
      设置 compilation debug="true" 以启用 ASPX 调试。否则，将此值设置为
      false 将提高此应用程序的运行时性能。
      设置 compilation debug="true" 以将调试符号 (.pdb 信息)
      插入到编译页中。因为这将创建执行起来
      较慢的大文件，所以应该只在调试时将该值设置为 true，而在所有其他时候都设置为
      false。有关更多信息，请参考有关
      调试 ASP.NET 文件的文档。
    -->
    <compilation defaultLanguage="c#" debug="true" targetFramework="4.0">
  </compilation>
  <!-- 自定义错误信息
      设置 customError 模式值可以控制应向
      用户显示用户友好错误信息而不是错误详细信息（包括堆栈跟踪信息）：

      "On" 始终显示自定义（友好的）信息
      "Off" 始终显示详细的 ASP.NET 错误信息。
      "RemoteOnly" 只对不在本地 Web 服务器上运行的
      用户显示自定义（友好的）信息。出于安全目的，建议使用此设置，以便
      不向远程客户端显示应用程序的详细信息。
    -->
    <customErrors mode="RemoteOnly"/>
  <!-- 身份验证
      此节设置应用程序的身份验证策略。可能的模式是 "Windows"、"Forms"、
      "Passport" 和 "None"
    -->
    <authentication mode="None"/>
  <!-- 应用程序级别跟踪记录
      应用程序级别跟踪在应用程序内为每一页启用跟踪日志输出。
      设置 trace enabled="true" 以启用应用程序跟踪记录。如果 pageOutput="true"，则
      跟踪信息将显示在每一页的底部。否则，可以通过从 Web 应用程序
      根浏览 "trace.axd" 页来查看
      应用程序跟踪日志。
    -->
    <trace enabled="false" requestLimit="10" pageOutput="false" traceMode="SortByTime" localOnly="true"/>
  <!-- 会话状态设置
      默认情况下，ASP.NET 使用 cookie 标识哪些请求属于特定的会话。
      如果 cookie 不可用，则可以通过将会话标识符添加到 URL 来跟踪会话。
      若要禁用 cookie，请设置 sessionState cookieless="true"。
    -->
    <sessionState mode="InProc" stateConnectionString="tcpip:127.0.0.1:42424" sqlConnectionString="data source=127.0.0.1;
      user id=sa;password=" cookieless="false" timeout="20"/>
  <!-- 全球化
      此节设置应用程序的全球化设置。
    -->
    <globalization requestEncoding="utf-8" responseEncoding="utf-8"/>
    <xhtmlConformance mode="Legacy"/>
    <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/></system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CSharp.CSharpCodeProvider, System, Version=4.0.0.0,
        Culture=neutral, PublicKeyToken=b77a5c561934e089" warningLevel="4" compilerOptions="/d:DEBUG;TRACE">
        <providerOption name="CompilerVersion" value="v4.0"/>
        <providerOption name="WarnAsError" value="false"/></compiler>
    </compilers></system.codedom>
</configuration>

```

APPENDIX 3:

```

namespace WMS
{
    /// <summary>
    /// _Default 的摘要说明。
    /// </summary>
    public partial class _Default : System.Web.UI.Page
    {

        protected void Page_Load(object sender, System.EventArgs e)
        {
            // 在此处放置用户代码以初始化页面
            if (Request["Menu"] == "Exit")
            {
                Session["uid"] = "";
                Session.RemoveAll();
                Response.Write("<script language=javascript>top.location.href='/'</script>");
                Response.End();
            }
        }

        protected void btn_login_Click(object sender, System.EventArgs e)
        {
            //从文件Web.config中读取连接字符串
            string strconn= ConfigurationSettings.AppSettings["dsn"];
            //连接本地计算机的HMS数据库
            //SqlConnection cn = new SqlConnection("Data Source=.\SQLExpress;Initial Catalog=HMS;Integrated Security=True");
            SqlConnection cn = new SqlConnection("Data Source=.\SQLExpress;Initial Catalog=HMS;Integrated Security=True");
            //SqlConnection cn = new SqlConnection("Data Source=data source=.\SQLEXPRESS;Integrated Security=SSPI;
            //AttachDBFilename=|DataDirectory|\\

            //Data Source=.\SQLExpress;AttachDbFilename=@quot;
            //C:\Documents and Settings\Administrator\My Documents\Visual Studio 2010\Projects\
            //WebApplication1\WebApplication1\App_Data\HMS_Data.MDF@quot;;Integrated Security=True

            //HMS_Data.MDF;User Instance=true");
            cn.Open();

            //构造SQL语句, 该语句在Users表中检查用户名和密码是否正确
            string mysql= "select * from Users where UID='"+tbx_uid.Text.Trim()+"' and UPassword='"+tbx_upassword.Text.Trim()+"'";
            //创建Command对象
            SqlCommand cm=new SqlCommand (mysql, cn);
            //执行ExecuteReader ()方法

            SqlDataReader dr=cm.ExecuteReader ();
            if(dr.Read ())
            {
                //保存当前用户名及用户权限
                Session["uid"]=dr["UID"].ToString ();

                //进入酒店总管界面
                Response.Redirect ("Admin.aspx");
            }
            else
            {
                Response.Write("<script language=javascript>alert('system warning: username or password is not right!');history.back();</script>");
                Response.End();
            }
            //关闭连接
            cn.Close();
        }
    }
}

```

APPENDIX 4:

```

public partial class CManage : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if (!IsPostBack) BindGrid();
    }
    public void BindGrid()
    { //从文件Web.config中读取连接字符串
        string strconn = ConfigurationSettings.AppSettings["dsn"];
        //连接本地计算机的HMS数据库
        SqlConnection cn = new SqlConnection(strconn);
        //创建SqlDataAdapter对象,调用存储过程
        SqlDataAdapter da = new SqlDataAdapter("CList", cn);
        //创建并填充DataSet
        DataSet ds = new DataSet();
        da.Fill(ds);
        dgd_usermanage.DataSource = ds;
        dgd_usermanage.DataBind();
        cn.Close();
    }

    public void DataGrid_Delete(Object sender, DataGridCommandEventArgs E)
    {
        //从文件Web.config中读取连接字符串
        string strconn = ConfigurationSettings.AppSettings["dsn"];
        //连接本地计算机的HMS数据库
        SqlConnection cn = new SqlConnection("Data Source=.\SQLExpress;Initial Catalog=HMS;Integrated Security=True");
        cn.Open();
        SqlCommand cm = new SqlCommand("CDelete", cn);
        cm.CommandType = CommandType.StoredProcedure;
        cm.Parameters.Add("@ID", SqlDbType.Int);
        //从DataGrid中取得更新内容
        //Cells [0]为UID列
        cm.Parameters["@ID"].Value = E.Item.Cells[0].Text.ToString();
        cm.ExecuteNonQuery();

        // ----- 处理最后一页只有一条记录, 如果删除出错的情况 -----
        if (dgd_usermanage.Items.Count == 1 && dgd_usermanage.CurrentPageIndex > 0)
        {
            dgd_usermanage.CurrentPageIndex--;
        }

        dgd_usermanage.EditItemIndex = -1;
        BindGrid();
    }

    public void DataGrid_Page(Object sender, DataGridPageChangedEventArgs E)
    {
        dgd_usermanage.CurrentPageIndex = E.NewPageIndex;
        BindGrid();
    }
}

```

APPENDIX 5:

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

    }

    protected void btnOperate_Click(object sender, EventArgs e)
    {
        try
        {
            string Sql = "backup database HMS to disk=" + Server.MapPath("/DataBack/data.bak") + "";
            string strconn = ConfigurationSettings.AppSettings["dsn"];
            //连接本地计算机的HMS数据库
            SqlConnection cn = new SqlConnection(strconn);
            cn.Open();
            //利用Command对象调用存储过程
            SqlCommand cm = new SqlCommand(Sql, cn);
            cm.ExecuteNonQuery();
            //关闭连接
            cn.Close();
            Response.Write("<script language=javascript>alert('system warning:backup database successful');location.href='main.aspx'</script>");
            Response.End();
        }
        catch(Exception ErrInfo)
        {
            Response.Write("<script language=javascript>alert('system warning: " + ErrInfo + "');history.back();</script>");
            Response.End();
        }
    }
}
```

APPENDIX 6:

```

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

    }

    protected void btn_submit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            //从文件Web.config中读取连接字符串
            string strconn = ConfigurationSettings.AppSettings["dsn"];
            //连接本地计算机的HMS数据库
            SqlConnection cn = new SqlConnection("Data Source=.\SQLExpress;Initial Catalog=HMS;Integrated Security=True");
            cn.Open();
            //利用Command对象调用存储过程
            SqlCommand cm = new SqlCommand("YLadd", cn);
            //将命令类型转为存储类型
            cm.CommandType = CommandType.StoredProcedure;
            //往存储过程中添加参数
            cm.Parameters.Add("@YName", SqlDbType.VarChar);
            cm.Parameters.Add("@YPrice", SqlDbType.Float);
            cm.Parameters.Add("@YNum", SqlDbType.Int);
            cm.Parameters.Add("@YTotalPrice", SqlDbType.Float);

            //给存储过程的参数赋值
            cm.Parameters["@YName"].Value = tbx_tname.Text.ToString();
            cm.Parameters["@YPrice"].Value = Convert.ToDouble(tbx_uname.Text);
            cm.Parameters["@YTotalPrice"].Value = Convert.ToDouble(tbx_uname.Text) * Convert.ToInt32(tbx_num.Text);
            cm.Parameters["@YNum"].Value = Convert.ToInt32(tbx_num.Text);
            cm.ExecuteNonQuery();
            //关闭连接
            cn.Close();
            Response.Redirect("YLmanage.aspx");
        }
    }

    protected void cv_Y_ServerValidate(object source, ServerValidateEventArgs args)
    {
        string strconn = ConfigurationSettings.AppSettings["dsn"];
        //连接本地计算机的HMS数据库
        SqlConnection cn = new SqlConnection(strconn);
        cn.Open();
        //构造SQL语句, 该语句在Users表中检查用户名是否存在
        string mysql = "select * from YuanLiaoInfo where YName=' " + tbx_tname.Text + "'";
        //创建Command对象
        SqlCommand cm = new SqlCommand(mysql, cn);
        //执行ExecuteReader ()方法
        SqlDataReader dr = cm.ExecuteReader();
        if (dr.Read())
        {
            args.IsValid = false; //用户名已被注册
        }
        else
        {
            args.IsValid = true; //用户名尚未被注册
        }
        //关闭连接
        cn.Close();
    }
}

```

APPENDIX 7/1:

```

public partial class OrderAdd : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        for (int i = 2007; i <= 2020; i++)
        {
            ddl_Y.Items.Add(new ListItem(i.ToString(), i.ToString()));
        }

        for (int i = 1; i <= 12; i++)
        {
            ddl_M.Items.Add(new ListItem(i.ToString(), i.ToString()));
        }

        for (int i = 1; i <= 31; i++)
        {
            ddl_D.Items.Add(new ListItem(i.ToString(), i.ToString()));
        }

        for (int i = 0; i <= 23; i++)
        {
            ddl_H.Items.Add(new ListItem(i.ToString(), i.ToString()));
        }

        for (int i = 0; i <= 59; i++)
        {
            ddl_S.Items.Add(new ListItem(i.ToString(), i.ToString()));
        }

        string strconn = ConfigurationSettings.AppSettings["dsn"];
        //连接本地计算机的HMS数据库
        SqlConnection cn = new SqlConnection("Data Source=.\SQLEXPRESS;Initial Catalog=HMS;Integrated Security=True");
        cn.Open();
        //利用Command对象调用存储过程
        SqlCommand cm = new SqlCommand("select * from CaiInfo order by ID", cn);
        SqlDataReader mdr = cm.ExecuteReader();
        while (mdr.Read())
        {
            chkbl.Items.Add(new ListItem(mdr["CName"].ToString(), mdr["CPriceB"].ToString() + "|" + mdr["CPriceC"].ToString()));
        }
        cn.Close();
    }

    protected void btn_submit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            if (chkbl.SelectedValue == "")
            {
                Response.Write("<script language=javascript>alert('system warning: please choose the food!');history.back();</script>");
                Response.End();
            }

            if (DateTime.Now.AddSeconds(30) > Convert.ToDateTime(ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value
                + "-" + ddl_D.SelectedItem.Value + " " + ddl_H.SelectedItem.Value + ":" + ddl_S.SelectedItem.Value + ":00"))
            {
                Response.Write("<script language=javascript>alert('system warning: please choose right booking time!');history.back();</script>");
                Response.End();
            }

            string[] strStr;

```

APPENDIX 7/2:

```

string strA = "";
string strB = "";

//从文件Web.config中读取连接字符串
string strconn = ConfigurationSettings.AppSettings["dsn"];
//连接本地计算机的MS数据库
SqlConnection cn = new SqlConnection(strconn);
cn.Open();
//利用Command对象调用存储过程
SqlCommand cm = new SqlCommand("OrderAdd", cn);
//将命令类型转为存储类型
cm.CommandType = CommandType.StoredProcedure;
//往存储过程中添加参数
cm.Parameters.Add("@T", SqlDbType.Int);
cm.Parameters.Add("@P", SqlDbType.Int);
cm.Parameters.Add("@C", SqlDbType.VarChar);
cm.Parameters.Add("@M", SqlDbType.Float);
cm.Parameters.Add("@IT", SqlDbType.Float);
cm.Parameters.Add("@Yue", SqlDbType.VarChar);
cm.Parameters.Add("@OT", SqlDbType.DateTime);

//给存储过程的参数付值
cm.Parameters["@T"].Value = Convert.ToInt32(tbx_TNum.Text);
cm.Parameters["@P"].Value = Convert.ToInt32(tbx_PNum.Text);

double strItems = 0;
double strMoney = 0;
foreach (ListItem RoleItem in chkbl.Items)
{
    if (RoleItem.Selected == true)
    {
        strStr = (RoleItem.Value).Split('|');
        strA = strStr[0];
        strB = strStr[1];

        strItems = strItems + Convert.ToDouble(strB);
        strMoney = strMoney + Convert.ToDouble(strA);
    }
    else
    {
        strItems = strItems + 0;
        strMoney = strMoney + 0;
    }
}

cm.Parameters["@M"].Value = strItems;

string strRoleItems = null;
foreach (ListItem RoleItem in chkbl.Items)
{
    if (RoleItem.Selected == true)
    {
        strRoleItems = strRoleItems + RoleItem.Text + "|";
    }
    else
    {
        strRoleItems = strRoleItems + "";
    }
}
// ----- 去掉最后一个|符号 -----
int intRoleItemsLen = strRoleItems.Length;

```

APPENDIX 7/3:

```
        strRoleItems = strRoleItems.Substring(0, intRoleItemsLen - 1);

        cm.Parameters["@TT"].Value = strMoney;
        cm.Parameters["@C"].Value = strRoleItems;
        cm.Parameters["@Yue"].Value = ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value;

        cm.Parameters["@OT"].Value = Convert.ToDateTime(ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value + "-" + ddl_D.
        cm.ExecuteNonQuery();
        //关闭连接
        cn.Close();
        Response.Redirect("Ordermanage.aspx");
    }
}

protected void cv_I_ServerValidate(object source, ServerValidateEventArgs args)
{
    string strconn = ConfigurationSettings.AppSettings["dsn"];
    //连接本地计算机的HMS数据库
    SqlConnection cn = new SqlConnection(strconn);
    cn.Open();
    //构造SQL语句, 该语句在Users表中检查用户名是否已存在
    string mysql = "select * from OrderInfo where TableNO=" + Convert.ToInt32(tbx_TNum.Text) + " and OrderTime=" + Convert.
        ToDateTime(ddl_Y.SelectedItem.Value + "-" + ddl_M.SelectedItem.Value + "-" + ddl_D.SelectedItem.Value + " " +
        |ddl_H.SelectedItem.Value + ":" + ddl_S.SelectedItem.Value + ":00") + """;
    //创建Command对象
    SqlCommand cm = new SqlCommand(mysql, cn);
    //执行ExecuteReader ()方法
    SqlDataReader dr = cm.ExecuteReader();
    if (dr.Read())
    {
        args.IsValid = false; //用户名已被注册
    }
    else
    {
        args.IsValid = true; //用户名尚未被注册
    }
    //关闭连接
    cn.Close();
}
}
```