



Masters Project Final Report

January 2017

Project Title	Online Restaurant Management & Food Ordering System		
Student Name	S. A. H. K. Subasinghe		
Registration No. & Index No.	14550583 2014/MIT/058		
Supervisor's Name	Dr. K. L. Jayaratne		
Please Circle the appropriate	Master's Program		Type
	MIT	MCS	Research Implementation

For Office Use ONLY

Online Restaurant Management & Food Ordering System

S. A. H. K. Subasinghe

2017



Online Restaurant Management & Food Ordering System

**A dissertation submitted for the Degree of Master of
Information Technology**

**S. A. H. K. Subasinghe
University of Colombo School of Computing
2017**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : S A H K Subasinghe

Registration Number : 2014/MIT/058

Index Number : 14550583

Signature:

Date:

This is to certify that this thesis is based on the work of Ms. S. A. H. K. Subasinghe under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by :

Supervisor Name : Dr. K L Jayaratne

Signature:

Date:

Signature:

Date:

Abstract

Typically in a restaurant food order process involves several steps for ordering the food where firstly customer starting from browsing the paper based menu and then inform to the waiter for ordering items. An alternative method for the customers is “Online food ordering System” using Web Based Application.

The online food ordering system is one of the latest services most of the restaurants in the world are adopting. Increased demand of restaurant-goers generated the need for much attention for the hospitality industry. Providing much option with ease of ordering and delivering is the need of the hours. Technological interference has become mandatory to improve the quality of the service and business in this industry.

The purpose of this project is to develop an online food ordering and restaurant management system for platinum pier restaurant at Mirage Colombo. Currently they are handling daily routine of their restaurant and food ordering process manually.

The proposed online food ordering system greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

The online food ordering system gives restaurants the ability to increase sales and expand their business by giving customers the facility to order food online. Also it simplifies the restaurant management process also.

The system was developed using the PHP and Java language while following Model-View-Controller model. Interfaces improved by using the JQuery, HTML and FusionChart were used to generate reports. The system runs on Apache Tomcat web server and MySQL for database in Windows environment.

According to the specified requirements, the system was designed and developed using methodologies and technologies mentioned above. The system has fulfilled its requirements and is successfully implemented.

Acknowledgment

Many people contributed their time, effort and ideas into completing this dissertation. I would like to thank them all.

First I would like to thank the Directors and all the staff members of the University Of Colombo School Of Computing for giving us all the support and all the studding materials.

This dissertation could not have been written without Dr. Lakshman Jayaratne, who not only served as my supervisor, but also encouraged and challenged me throughout my academic program.

I would like to give my special thanks to Mr. Danushka Amila Kumara, Assistant Manager of Mirage Colombo Hotel who gave me extensive help and support during the collecting system requirements and guidance that enabled me to develop an understanding of the domain, as well as other staff members who invested time for providing me with necessary additional details.

And I would like to offer my gratitude for all those who supported me in many ways during the completion of the project. Finally and most especially to my family and friends that I owe them for their encouragement to complete this project successfully. I thank them all.

Contents

List of Figures	ix
List of Tables	xi
List of Abbreviations.....	xii
Chapter 1: Introduction.....	1
1.1 Overview	1
1.2 Motivation for the project	1
1.3 Objectives of the project	1
1.4 Scope of the project	2
1.5 Benefits from the system	4
1.6 Structure of the dissertation.....	4
Chapter 2: Background and Literature Review.....	6
2.1 Background.....	6
2.2 Self-Ordering Systems	6
2.3 literature Review	7
Summary	10
Chapter 3: Analysis and Design.....	11
3.1 Disadvantages of manual process	11
3.2 Proposed System.....	11
3.3 Requirements Gathering Methods.....	12
3.4 Functional Requirements	14
3.5 Nonfunctional Requirements	15
3.6 Similar Systems Available	16
3.7 Alternative Solution Evaluation.....	17
3.8 Initial use case Diagram.....	20
3.8.1 Use case narratives	20
3.9 Description of the system users	21
3.10 Entity Relationship Diagram.....	22
3.12 Activity Diagrams.....	24
3.13 State Diagrams.....	25
3.14 Design Decisions	25
3.15 System Interfaces.....	26
Chapter 4: Implementation.....	29
4.1 Software Background	29
4.1.1 System Implementation	30

4.2 Usage of Design Patterns	31
4.3 Development Tools	31
4.4 User Interface Design.....	32
4.5 Reusable Codes.....	33
4.6 Program List.....	33
Chapter 5: Evolution and Testing.....	36
5.1 Test Plan.....	36
5.2 Test Plan of the System	40
5.3 Test Cases.....	40
5.4 User Evaluation	42
Chapter 6: Conclusion and Future Work	44
6.1 Project Assessment.....	44
6.2 Problems Encountered.....	45
6.3 Lessons Learned	46
6.4 Achievements	46
6.5 Future Developments	47
References	48
Appendix A – System Documentation	49
Appendix B - User Evaluation Form	51
Appendix C - Use case narratives	52
Appendix D – Test cases	56
Appendix E – System interfaces.....	62
Appendix F – Program List.....	65

List of Figures

Figure 1.1 - System model	3
Figure 3.1 – Use case diagram	12
Figure 3.2 – ER diagram	14
Figure 3.3 – Sequence diagram for reset user password	16
Figure 3.4 - Activity diagram for user login	17
Figure 3.5 - State diagram for the account object	18
Figure 3.6- home page	26
Figure 3.7 – food zone.....	27
Figure 3.8 – admin home page.....	27
Figure 3.9 - report generation.....	28
Figure 4.1- config.php	33
Figure 4.2 - locale.php.....	34
Figure 4.3 - registration form.....	35
Figure 4.4-customer login form validation.....	35
Figure 5.1-stat of user satisfaction level.....	42
Figure 5.2 – stats of user evaluation of system performance.....	42
Figure 5.3- customer satisfaction vs. user friendliness.....	43
Figure E.1 – shopping cart.....	62
Figure E.2-order history.....	62
Figure E.3-category update.....	63
Figure E.4-add/remove food items.....	63
Figure E.5-veiw customer list.....	64
Figure F.1-validation of admin login.....	65

Figure F.2-login failed.....	65
Figure F.3-register form validation.....	66
Figure F.4-validation of table reservation.....	67
Figure F.5-order confirmation.....	68

List of Tables

Table 3. 1 – Use case narrative for admin login	12
Table 3.2 - Database Privilege	19
Table 5.1 – unit test for customer log in.....	37
Table 5.2 – unit test for add new food category.....	37
Table 5.3 – Test plan	40
Table 5.4 - Test case for administrator log in	40
Table 5.5 - test case for Customer create orders	41
Table A. 1 Recommended Hardware Requirements	49
Table A. 2 Recommended Software Requirements	49
Table C: 1 Use case narrative for customer log in	52
Table C: 2 Use case narrative for create customer account	52
Table C: 3 Use case narrative for place order	54
Table C: 4 Use case narrative for generate reports	54
Table C: 5 Use case narrative for customer pay bill	54
Table C: 6 Use case narrative for manage user accounts	55
Table D. 1 Test case for customer payments	56
Table D. 2 Test case for customer create account	58
Table D. 3 Test case for Manage user accounts.....	61

List of Abbreviations

CMS	Content Management System
ER	Entity Relationship
FIWOS	Fountains Integrated Wireless Ordering System
JSP	Java Server Pages
My SQL	My Structured Query Language
OS	Operating System
PDA	Personal Digital Assistance
PHP	Hypertext Preprocessor
SMS	Short Message Service
UML	Unified Modeling Language
WOS	Wireless Ordering System

Chapter 1: Introduction

1.1 Overview

Mirage Colombo is a hotel located in Colombo 6. They are handling daily routine of their restaurant and food ordering process manually. Customers cannot order food online in Mirage Colombo hotel. In case of the food take away or dining there, customer has to visit or call and order. Table reservation is also the same. This project is based on automating the above mentioned processes.

Online ordering system that proposing here, greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

1.2 Motivation for the project

The motivation for designing this application came because my spouse is involved in the hotel industry and I personally do not like waiting for long in the store or to have to call store to place an order especially during the peak lunch or dinner hours. Moreover, I value learning about Programming languages as well as seeing how powerful and dynamic they are when it comes to web designing and applications.

1.3 Objectives of the project

- The main objective of the project is to automate the manual work of the restaurant of Mirage Colombo.
- Improve the customer relationship management.
- Reduce the waiting time of the customers & improve the customer satisfaction.
- Avoid long queues at the counter.
- Introduce a new restaurant management & online food ordering system which can be implemented in any restaurant.

1.4 Scope of the project

Mirage Colombo is a hotel located in Colombo. In this Hotel they are having a la carte menu for lunch on every day and for dinner from Monday to Thursday. Also having buffet dinner from Friday to Sunday.

What I propose here is an online ordering system (a web based system) for Mirage Colombo. But this system is applicable in any food delivery industry. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and the restaurant. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order. Also customers can reserve tables through this system if they are vising for dining.

This system also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated. Once an order is placed on the webpage, it is entered into the database and then retrieved, in pretty much real-time, by a desktop application on the restaurant's end. Within this application, all items in the order are displayed, along with their corresponding options and delivery details, in a concise and easy to read manner. This allows restaurant employees to quickly go through the orders as they are placed and produce the necessary items with minimal delay and confusion.

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.

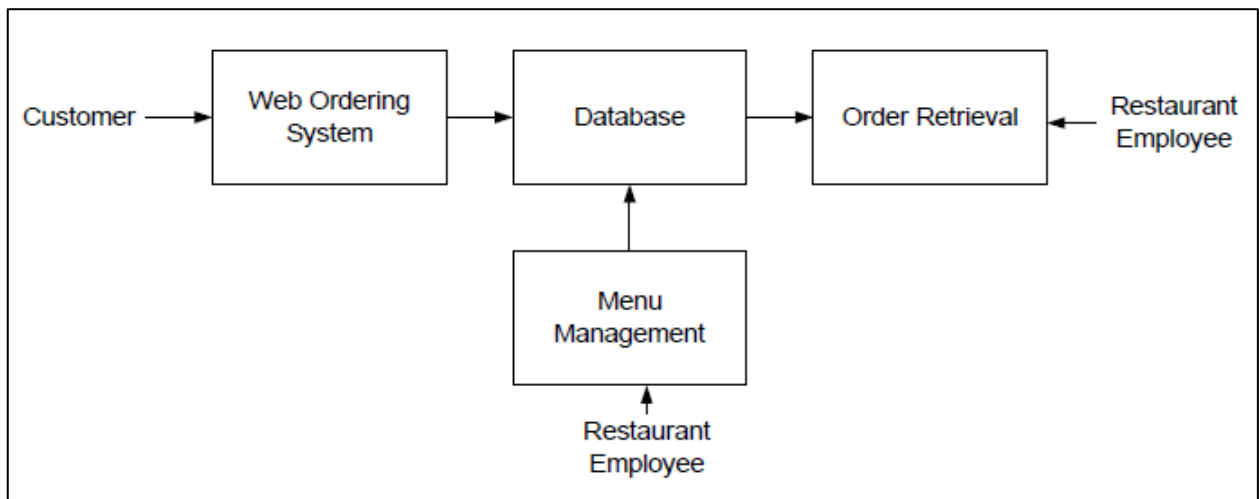


Figure 1.1 – System model

Users of the web ordering system, namely restaurant customers, must be provided the following functionality:

- Create an account
- Manage their account
- Log in to the system
- Navigate the restaurant’s menu
- Select an item from the menu
- Add an item to their current order
- Review their current order
- Remove an item/remove all items from their current order
- Provide delivery and payment details
- Place an order or reserve tables
- Receive confirmation in the form of an order number

The menu management system will be available only to restaurant employees and will, as the name suggests, allow them to manage the menu that is displayed to users of the web ordering system. The functions afforded by the menu management system provide user with the ability to, using a graphical interface:

- Add a new/update/delete food category to/from the menu
- Add a new/update/delete food item to/from the menu
- Add a new/update/delete option for a given food item
- Update price for a given food item

- Update default options for a given food item
- Update additional information (description, photo, etc.) for a given food item

Of the three components, the order retrieval system is functionally the simplest. Like the menu management system, it is designed to be used only by restaurant employees, and provides the following functions:

- Retrieve new orders from the database
- Display the orders in an easily readable, graphical way
- Mark an order as having been processed and remove it from the list of active orders

1.5 Benefits from the system

- Minimize the number of employees at the back of the counter.
- Reduce manual paper work.
- The system will help to reduce labor cost involved.
- Less probable to make mistake, since it is a machine.
- Avoid long queues at the counter due to the speed of execution and number of optimum screens to accommodate the maximum throughput.
- Efficiency and effective service to the customers.
- Ensured only authorized people have access to the relevant information.
- Security of the data information.
- Support for the business process decisions.
- Control privacy and confidentiality and stop inappropriate disclosure of information

1.6 Structure of the dissertation

Chapter one: The first chapter will simply describe overview of the system and background of the current process and it includes the introduction of the project, and the introduction to the problem, the scope, objectives of the project finally structure of the thesis.

Chapter two: This chapter describes the background of the system. It describes similar systems that available and alternative technologies available relevant to the system. It will include essential background information with references and benefits and weaknesses of those alternatives.

Chapter three: Third chapter describes is the Analysis and Design of the system.

In the analysis, it will include description about the current system as well as the user requirements, functional areas and non-functional areas.

In design, it will describe the details about the how system designed. It is contains relevant UML diagrams also.

Chapter four: The fourth chapter will describe the implementation plan of the system.

This will describe the plan of implementing the system and what we have been achieved up to date through results that already have accomplished.

Furthermore this chapter covers content of the changed work from the original proposal with reasons. And problem faced when implementing the system, the scope of the project compared with the time plan as well as result we have achieved. Furthermore it will describe test plan that was used to verify and validate system, plus small description of the effect of errors.

Chapter 2: Background and Literature Review

This chapter gives a brief introduction about the project background. Here describes about traditional meter reading calculations methods and existing alternate bill calculation methods and technologies that used.

2.1 Background

In today's age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders while offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system, including the inconvenience of the customer needing to have a physical copy of the menu, lack of a visual confirmation that the order was placed correctly, and the necessity for the restaurant to have an employee answering the phone and taking orders.

Online food ordering has become a mandatory feature of a restaurants these days. The customers of today are not only attracted because placing an order online is very convenient but also because they have visibility into the items offered, price and extremely simplified navigation for the order.

Web based online food ordering systems are using by many restaurants all over the world. Some of those restaurants are McDonald's, Pizza hut, Dominos, Sponge etc. Online food ordering is possible via mobile applications also now a days and it is the new trend. Most of the above mentioned restaurants also having mobile applications.

2.2 Self-Ordering Systems

An ordering system is referred to as a set of detail methods that is being used in handling the ordering process. Food ordering can be computerized or done manually. Thus helps the customer to order their food themselves which is known as the customer self-ordering system.

The customer self-ordering system can be defined as a computerized system that is being used by customers to place their own orders in the restaurant and allow the orders to be tracked, in order to prepare and deliver the food to the customers.

Self-ordering in restaurant industry refers to the restaurant taking orders from customers through applying various types of technologies such as internet and many others. Self-ordering is successful when it is applied at restaurants in many other countries. The usage of the self-ordering technology is proven to benefit most of the investors.

Most of the people hate waiting for an order. Therefore, they prefer self-ordering technology, which can be in form of text messaging and the internet. Usually, the customer prefers self-ordering because of speed and convenience in making order and transaction while minimize the miscommunication.

2.3 literature Review

In the pass decades, the rapid growing of network and wireless technology did a great impact for how people communicate with each and other remotely. At the same time, this technology also leads different kind industries to change their entire management aspect. Food and Beverage industry is one of the industries in the market that apply these technologies into their business processes that assist them to be much more convenience and efficient.

Various systems were established in market for the purpose of food ordering. Some of the existing systems are explained below.

Full Service Restaurant

Tradition food order process used in most full-service restaurants starting when a waiter brought the guests the paper-based menu, and then waiting for the guests to choose items from the menu and inform the waiter the order items. One of the most widely used food ordering system is the conventional paper based system. In this system all records are stored on paper. The main drawback of this system is papers can get easily lost or damaged. There is also wastage of money, time and paper. Paper-based systems do not provide any form of dynamicity. Even a small change requires the re-print of entire menu-card. Also large amount of human efforts are required, this system is not work properly because it has some error and from a customer's point of view it is time consuming.

Self Service Restaurant

This process required the guests to place order at the service counter in the restaurant. The guests shall have decision in advance, before presented at the counter, of which menu items to order. Menu catalog is mostly presented as posters placing behind the order counter.

Automated Food Ordering System

In order to reduce service cost and enhance customer experiences, few restaurants have invested in the service automation system. The automation system used to capture the food order from guests ranged in many forms but mostly comprise of an electronic device with a screen presenting the menu and accept user's input for order placing. First waiter takes the order from customer. After taking the order, waiter should enter that order in system where PC was set up. At the kitchen information was displayed on screen. The kitchen staff would then prepare the dishes according to order and after completion of order they would inform to waiter, who collected and delivered the dishes to the respective tables. The system was also informing the waiter about the availability of a dish. If a certain dish was not available then waiter was able to ask for changes or even cancel a customer's order. After serving the order, bill was generated at the cash counter as per customer order. The management had full authority to access all details of the customer which are fed into the system.

With the improvement in the computer and communication technology, various systems were launched in market for the purpose of computerization of the food ordering system. Some of the existing systems are mentioned below:

Personal Digital Assistants (PDA's) Based System

A number of wireless systems like WOS, i-menu, FIWOS were developed when new technologies and approaches being introduced to automate the food ordering process. All the above systems were PDA- based. The feature of PDA systems was that customers or waiters key in ordering process. Using wireless technology there was easy communication between the PDA's and server. But PDA based system also had several drawbacks. PDA based system increased the restaurants expenditures. PDA systems also did not provide any real time feedback from customers. Menu cards in the PDA's were not attractive and uninformative as it did not support images.

Multi-touch Technology

Multi-touch technology is an advance version to the existing touch technology where user has authority to control and perform operations concurrently on the electronic visual displays using multiple fingers inputs. Large displays such as from the tabletop and the wall screens are deemed to be essentials for information visualization purposes when dealing with multiple users sharing the same display. It is reported that the social interaction is highly improved among users using a shared display and input. But there are certain limitations of the multi-touchable restaurant management systems. Touch screens available in the market are of capacitive, resistive types which are very costly. Limitations of capacitive touch screen are not able to operate with stylus until it is of conductive material. One more disadvantage of capacitive touch screen is it is expensive, offers less durability and short life. The drawbacks of resistive touch screen include its inability to support multi-touch gestures, its poor visibility in direct sunlight and its less durability. The technology can be susceptible to data-noise, it may be affected by large amounts of dirt and dust in the environment.

Mobile apps

Now a days mobile applications has become a main feature of any kind of a business. Since all the people are using smart devices now a days they are more likely to get done most of the works through that. When comes to the hotel industry people want to have order food on the go, while in a meeting etc. This is where mobile apps come to play. But when using mobile apps it is very important to consider about secure online payment. So when providing a mobile app to the customers, should consider more about security since most of the frauds are doing using smart devices these days.

To overcome the limitations of above systems, solution is a customizable online food ordering system using web based application. The main objective of this project is to develop an online food ordering system which can be used in any restaurant.

The entire system can be cost effective and significant amounts of time and money can be saved by implementing it. Human errors in billing and taking orders would be totally eliminated, also sales and the revenue collection will increase.

Summary

Proposed system is about automating the manual process of the restaurant and provide efficient and effective service to the customers. This is a web based system and this is applicable in any food delivery industry.

Chapter 3: Analysis and Design

Chapter three describes brief description of the disadvantages of manual process and about purposed system and main functions that addressed by it. It also describes nonfunctional requirements, hardware and software requirements likewise methods and tools that used. Furthermore it represents initial UML diagram techniques that used and description of the system users as well as system reports that generated for them.

3.1 Disadvantages of manual process

- Customers have to visit or call to order food or to reserve table.
- Long queues at the counter.
- High waiting time of customer and decrease customer satisfaction.
- More labor cost.
- More paper work.
- Less security of information.
- Difficulty in report generation.
- Difficulty in using the data for data mining and decision making.
- Less accuracy of data.

3.2 Proposed System

Propose system is an online ordering system (a web based system). But this system is applicable in any food delivery industry. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and the restaurant. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order. Also customers can reserve tables through this system if they are vising for dining.

Benefits:

- Minimize the number of employees at the back of the counter.
- Reduce manual paper work.

- The system will help to reduce labor cost involved.
- Less probable to make mistake, since it is a machine.
- Avoid long queues at the counter due to the speed of execution and number of optimum screens to accommodate the maximum throughput.
- Efficiency and effective service to the customers.
- Ensured only authorized people have access to the relevant information.
- Security of the data information.
- Support for the business process decisions.
- Control privacy and confidentiality and stop inappropriate disclosure of information.

3.3 Requirements Gathering Methods

There are various types of methods available for requirement gathering such as Interviews, Questionnaires, User Observation and Analyzing Existing Documents etc. Methods used for requirement gathering of proposed system are explained below.

Interviews

Interviews can be separate in to two methods as One-on-one and group interviews. One-on-one interviews are the most common technique for gathering requirements, as well as one of the primary sources of requirements. The analyst should identify stakeholders to be interviewed. These can be users who interact with the current or new system, management, project financiers or anyone else that would be involved in the system.

Group interviews are similar to one-on-one interview, except there is more than one person being interviewed. Group interviews work well when the interviewees are at the same level or position. More thoughts and discussion can be generated, as someone in the group may state or suggest an idea that may have been overlooked by others, which in turn can lead to a discussion or provide more information on a particular issue. The interviewer can gauge which issues are more generally agreed upon, and which are which issues differ. A major disadvantage can be scheduling the interview. When more than one person are involved, it may be difficult, or become time consuming.

For the proposed system One-on-one interviews conducted with restaurant staff and some other employees in the Mirage Colombo hotel to get the information regarding current manual system and also to gather requirements for the proposed system. Group interviews conducted with management level employees of the hotel to gather requirements related to management level, non-functional requirements and also to collect information on current manual process. Also interviewed few customers also from platinum pier restaurant to collect information for the proposed system.

Analyzing Existing Documents

Analyzing existing documents can prove to be a useful technique in requirement gathering, on its own as well using it to supplement other techniques. Reviewing the current process and documentation can help the analyst understand the business, or system, and its current situation. Existing documentation will provide the analyst the titles and names of stakeholders who are involved with the system. This will help the analyst formulate questions for interviews or questionnaires to ask of stakeholders, in order to gain additional requirements.

Documents related to current manual process such as menu cards, payment recipes, reservation forms, documents related to order details etc analyzed to gather information and requirements for the proposed system.

User Observation

To get a better understanding of a user in their current working environment, the analyst may observe the user themselves. User observation is helpful in assisting the analyst by getting a full grasp of how the user interacts with the system. Sometimes stakeholders may find it difficult in explaining what exactly what their tasks consists of and what their requirements may be, observing the user in cases like these will help provide the requirements. User observation may also be useful in validating data that had been previously collected.

User observation done for the restaurant staff of Mirage Colombo hotel to get a better understanding of the current manual system and to gather information for the proposed system also. This was very useful to identify what functionalities and features should be available in the proposed system for the restaurant staff.

3.4 Functional Requirements

The structure of the system can be divided into three main logical components:

- Web Ordering System- provides the functionality for customers to place their order and supply necessary details.
- Menu Management-allows the restaurant to control what can be ordered by the customers
- Order Retrieval System-This is a final logical component. Allows restaurant to keep track of all orders placed. This component takes care of order retrieving and displaying order information.

Web Ordering System Module

This module provides the functionality for customers to place their order and supply necessary details. Users of the system, namely restaurant customers, must be provided the following functionality:

- Create an account.
- Manage their account.
- Log in to the system.
- Navigate the restaurant's menu.
- Select an item from the menu.
- Add an item to their current order.
- Review their current order.
- Remove an item/remove all items from their current order.
- Provide payment details.
- Place an order.
- Receive confirmation in the form of an order number.
- View order placed.

Menu Management System Module

This module provides functionality for the power user-Administrator only. It will not be available to any other users of the system like Restaurant Employees or Customers. Using a graphical interface, it will allow an Admin to manage the menu that is displayed to users of the Web ordering system:

- Add/update/delete food category to/from the menu.
- Add /update/delete food item to/from the menu.
- Update price for a given food item.
- Update additional information (description, photo, etc.) for a given food item.

Before customers can actually use this system, functionality provided by this component will have to be configured first. Once the initial configuration is done, this will be the least likely used component as menu updates are mostly seasonal and do not occur frequently.

Order Retrieval System Module

This is the simplest module out of all three modules. It is designed to be used only by restaurant employees, and provides the following functions:

- Retrieve new orders from the database.
- Display the orders in an easily readable, graphical way.

3.5 Nonfunctional Requirements

Usability

System should address the requirements of managing the web based system efficiency, managing customer orders and their payments. It should be produce relevant information about the each and relevant food and beverages so they could place the orders depend on that. System should make better performance at peak time.

Availability

System is a web based software, it is located in web server so any user can logged into the system anywhere in the world at any time(24 X 7 availability.), and it should keep backups on database at least monthly wise or weekly wise and system needs to secure access of confidential data (user's details).

Access Control

System should be properly access control by administers and customers. Administer accounts handle by super admin and customer accounts handle by administers.

Modifiable

Only administer can change menu items, relevant images etc. Those details can view customer as a read only view.

Platform compatibility

System is a browser based software. So it is not platform dependent, for access software it just uses web browser with any operating system.

Flexibility

System should able to be flexible to manage any kind of work load. It should be able to customize the SMS notifications and handle customer orders, and this will be highly desirable for future extension.

3.6 Similar Systems Available

McDonald's, the largest fast food chain in the world and the brand most associated with the term "fast food" was founded as a barbeque drive-in in 1940 by Dick and Mac. McDelivery is the McDonald's online food ordering service that delivers food to the customer's door. The service was introduced in parts of the United States beginning in 1993 and is available in many Asian, Middle Eastern and Latin American countries. In some countries, delivery is available 24 hours a day, and in at least one location, the service is free with a minimum order. McDelivery service started in Sri Lanka in 2014. Customers can order food online by using the menu given in the website and payment can be done via credit or debit cards through the web site. After the order completed within promised time order will be delivered to the given address using motorcycle couriers. One limitation here is customers cannot place an order less than Rs.600.

Brothers Dan and Frank Carney opened the world's first Pizza Hut restaurant in 1985 in Wichita USA. Since its inception in 1993, Pizza Hut has fast become a household name across Sri Lanka. With its first restaurant at Union Place, Colombo 2, Pizza Hut became the first international restaurant chain to begin operations in Sri Lanka. In 1999, Pizza Hut ventured into the home delivery service setting yet another innovative milestone in Sri Lanka. With the launch of its island wide delivery hotline 011 2 729729, Pizza Hut delivery rose swiftly to provide 50% of the restaurant's net sales. In August 1994 SCO and Pizza Hut announced PizzaNet, a pilot

program that enables computer users, for the first time, to electronically order pizza delivery from their local Pizza Hut restaurant via the worldwide Internet. PizzaNet was based on the first commercially licensed and bundled Internet Operating System, SCO Global Access. Pizza hut online ordering system is established in Sri Lanka also and it provides the newest and convenient features among the online food ordering systems used in various fast food restaurants. Also pizza hut has launched mobile app for online ordering.

Sponge, a restaurant located in Colombo has been successfully standing apart from the competitors because of applying online self-service ordering and the payment concepts. The system used in Sponge allows the customers make an order through their web site, provide billing and delivery address and pay online using PayPal, debit or credit cards. The system also provides database for generate the management reports, perform analysis as well as allows the menu to be updated instantly.

An online food ordering system is not a new system type. In fact, some of the software companies develop and commercialize the system. However, some restaurants still prefer to develop the food ordering system themselves due to cost and customization issues. Same as Mirage Colombo hotel also wanted to newly create an online food ordering system for their restaurant rather than buying and customizing a system due to limitations in those commonly available customizable systems. Main reason for that is mostly available systems are for fast food restaurants and platinum pier is not a fast food restaurant. Another point is they wanted to handle the table reservation also online for those who wanted to visit for dining there.

3.7 Alternative Solution Evaluation

Develop using ASP.NET technology

System can be developed using asp.net, it is a strong and powerful programming language. ASP.NET is MVC. It is a .NET implementation of the model-view-controller concept, which basically means developers have separate layers for data handling, presentation and business logic. It will result in much more web-style development, with cleaner separation of code and cleaner output

Then again it has some drawbacks such as developing tools are not available for free as well as it have higher write and retrieve performance in MS SQL server rather than oracle database. ASP. NET limits customizing interfaces (such as developing own JavaScript solutions,

advanced designs, etc.) and leaves with a very unmanageable HTML output, if someone want to work with it after it has been generated.

Develop using a Content Management System

A Content Management System is a web application that provides all the necessary tools to manage online content and users, or more specifically, a server-side software that facilitates the creation and maintenance of websites. There are many CMS available and some popular CMS include WordPress, Joomla and Drupal.

Probably one of the most notable perks of having a CMS is that users don't need extensive technical expertise to generate webpages, to upload or modify content. CMSs feature an administration area that allows users to upload or edit their content as needed. This admin area is accessible through a web-browser, which means that no client software installation is needed and an internet connection is all you need to start working on your website. A CMS essentially eliminates the need for a webmaster and allows users to take matters into their own hands.

Using a CMS can drastically speed up the process of developing a site. Most systems allow the implementation of functionalities (forms, polls, quizzes, event calendars, etc.) that would otherwise be extremely hard to achieve. CMSs are user-friendly and open source CMSs are free to download and install. Implementing an out-of-the-box CMS is definitely less costly than a custom one. Also, after users have learned some of the ins and outs of the CMS, they will be able to handle the basic functions on their own, and this reduces the costs of outside technical assistance as well.

There are some disadvantages such as hidden implementation costs for perfecting the system for users' needs. Some CMSs (e.g. Joomla) can put a load on your server resources. If users have an aversion to learning the basics of using a CMS, the usage of a CMS will not be as efficient. This means that hiring experienced staff or contracting technical help might be necessary. For community-based CMS systems, direct support may not be easy to find, and users have to rely on user forums and existing documentation. In the wrong hands, an unmaintained and outdated CMS can do a lot of harm to a website and lead to serious security vulnerabilities.

Develop as a Desktop Application

System has to be shared with several users in different locations. So another option was building a desktop application that users can connect through a network

Desktop applications remain reliable and trustworthy. Desktop applications are restricted to a physical location hence have usability constraint. These applications do not rely on an Internet connection. One of the major drawback in desktop application is one cannot access this desktop type of applications if they do not have a computer within reach. However web applications are exposed to more security threats than desktop applications. One can have a full control over the standalone applications and protect it from numerous vulnerabilities.

As an advantage web applications need to be installed only once whereas desktop applications are to be installed separately on each and every computer.

After considering the available solutions for the development of the proposed system, decided to make it web applications because users capable to access the application from any location using the internet and it offer easy accessibility to users from anywhere at any time. It will develop using PHP and servlet because of the advantage of PHP and java language; it is type safe, much faster and more reliable in deployment.

Web development is standards based and not required to use proprietary software it has fast development cycles. JavaScript, HTML, and CSS make it extremely helpful to build this application quickly.

3.8 Initial use case Diagram

The below figure (Figure 3.1) shows the use case diagram for the system. Three main actors found, that actors directly uses the system, below diagram shows actors and their main processors related to the system.

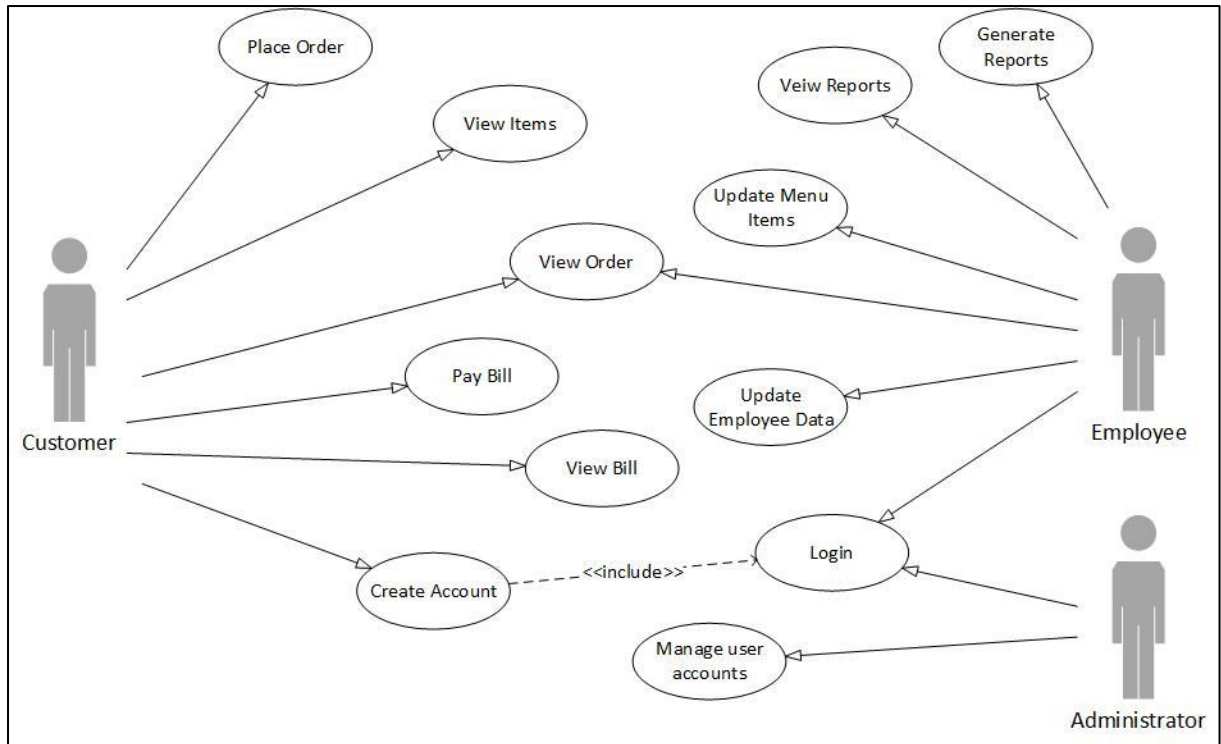


Figure 3.1 – Use case Diagram

3.8.1 Use case narratives

Table 3.1 shows the use case description for Administrator log in to the system.

Table 3. 1 – Use case narrative for admin login

ID: U01	Actors: user Administrator Title: Log in
Description:	Relevant admin comes to log in page Logged in using his user name and password.
Preconditions:	Administrator has a valid user name and password. User in log in page.
Main Flows:	Enter a valid user name. Enter valid password. User logged into the system.

Post Conditions:	Log in successfully and user redirect to home page
Alternative paths:	User enters invalid user name then an error message will be displayed.
	User enters an invalid password then error message will be displayed.
	User enters only user name then error message will be displayed.
	User enters only password then error message will be displayed.
	Administrator should re-enter log in details or terminate the process.

Refer Appendix C for more use case narratives.

3.9 Description of the system users

Customer :

One of the main user of the system. These users will use system for manage their personal details, to place orders, to make payments, view bill information and view order details.

Main functions

- Place orders
- Make payments
- View orders & bills
- Update user information

Employee :

Persons who have authority to manage item details, order details, customers and payments.

Main functions

- Manage items
- Mange order details
- Generate and view reports
- Manage payment details

Administrator :

Persons who have authority to manage user accounts and access permissions.

Main functions

- All administer functions
- Manage users & access permissions

3.10 Entity Relationship Diagram

It was decided to use OOP methodology to develop a system, but because of identifying the requirements of the system and to identify database tables, decided to draw ER diagram to get a clear overview of the proposed system. Figure 3.3 describes the relationship between each and every tables of the system.

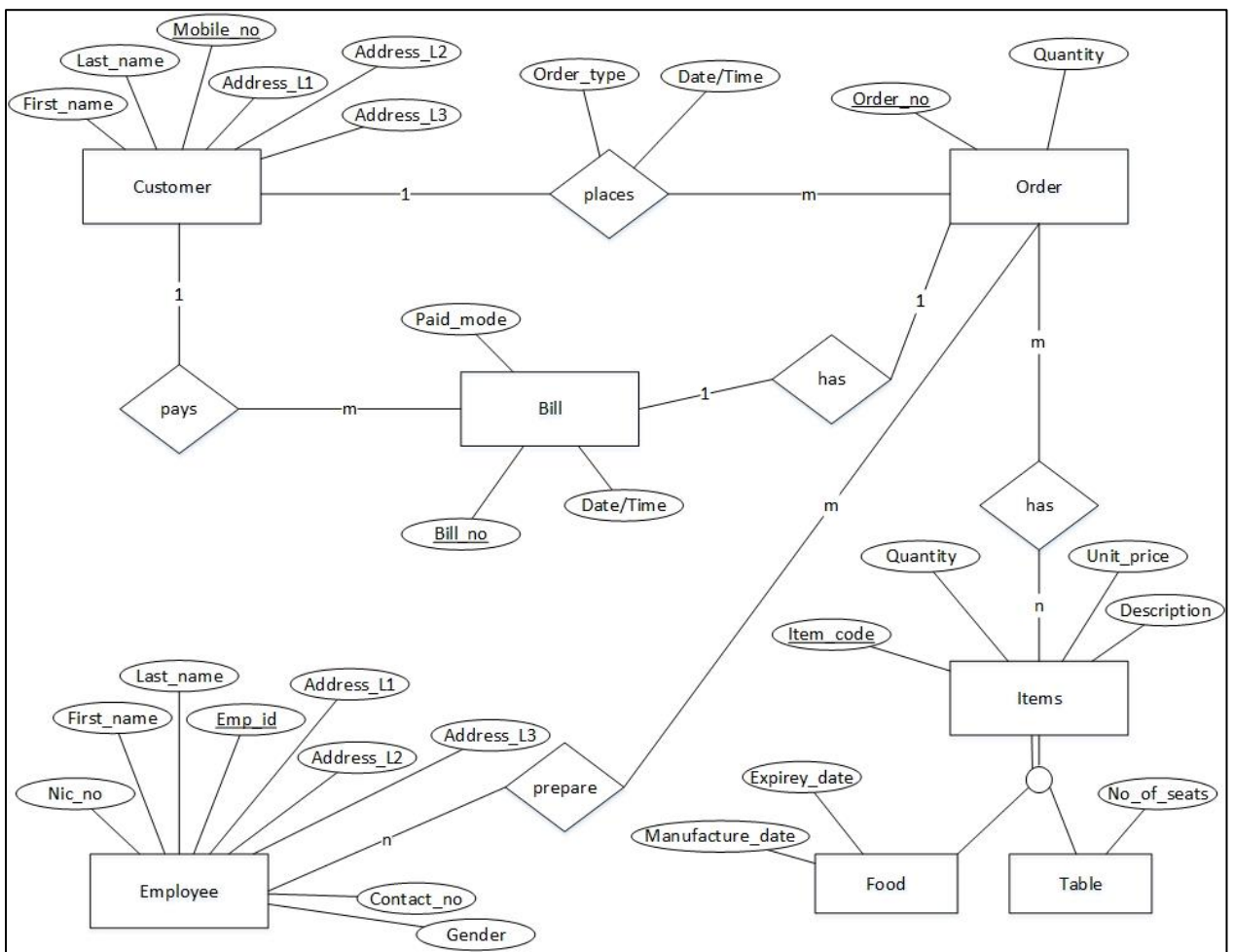


Figure 3.2 – ER diagram

3.11 Sequence diagrams

Following diagram (figure 3.3) shows the sequence diagrams for reset user password.

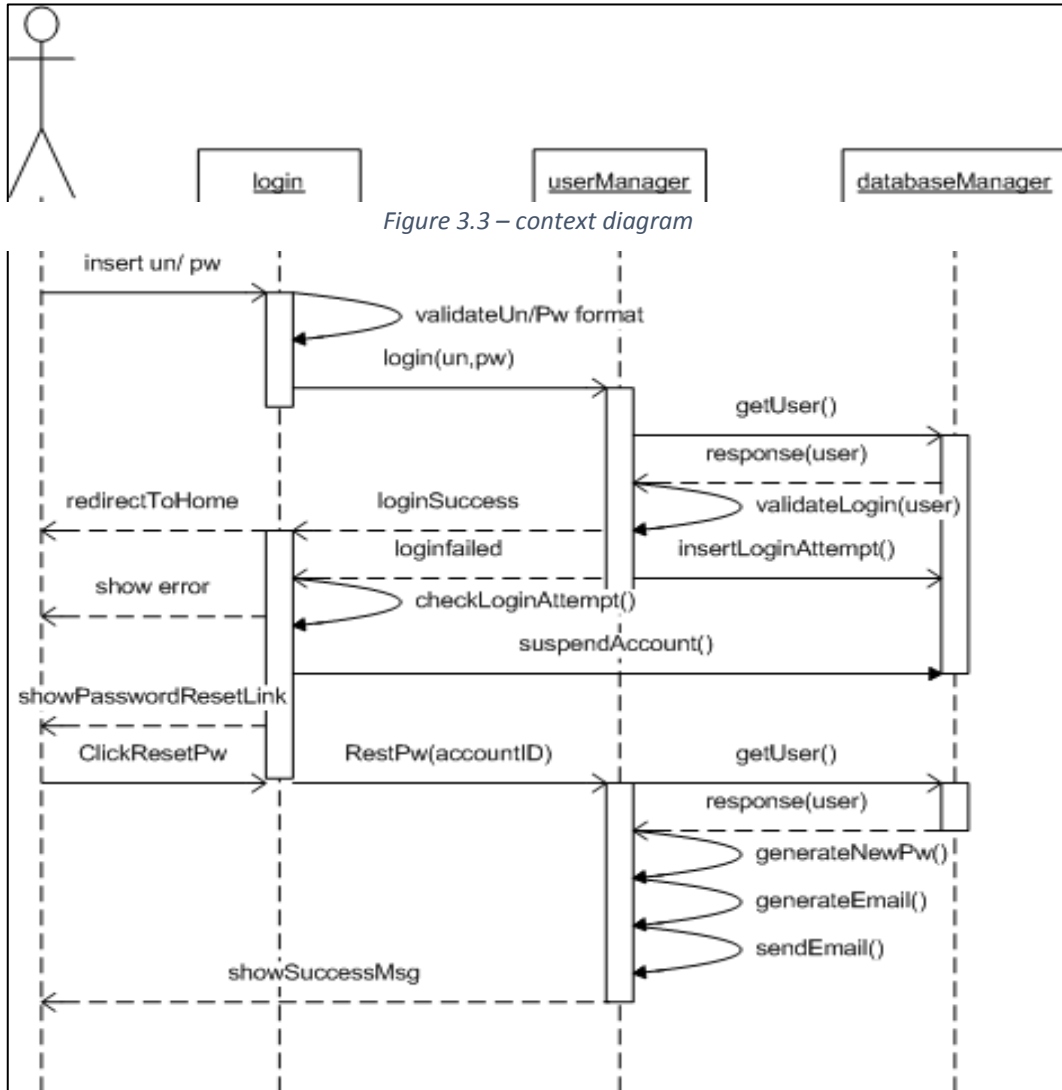


Figure 3.3 – Sequence diagram for reset user password

3.12 Activity Diagrams

The following Activity diagram (Figure 3.4) describes functionality that occurred when a user try to log-in.

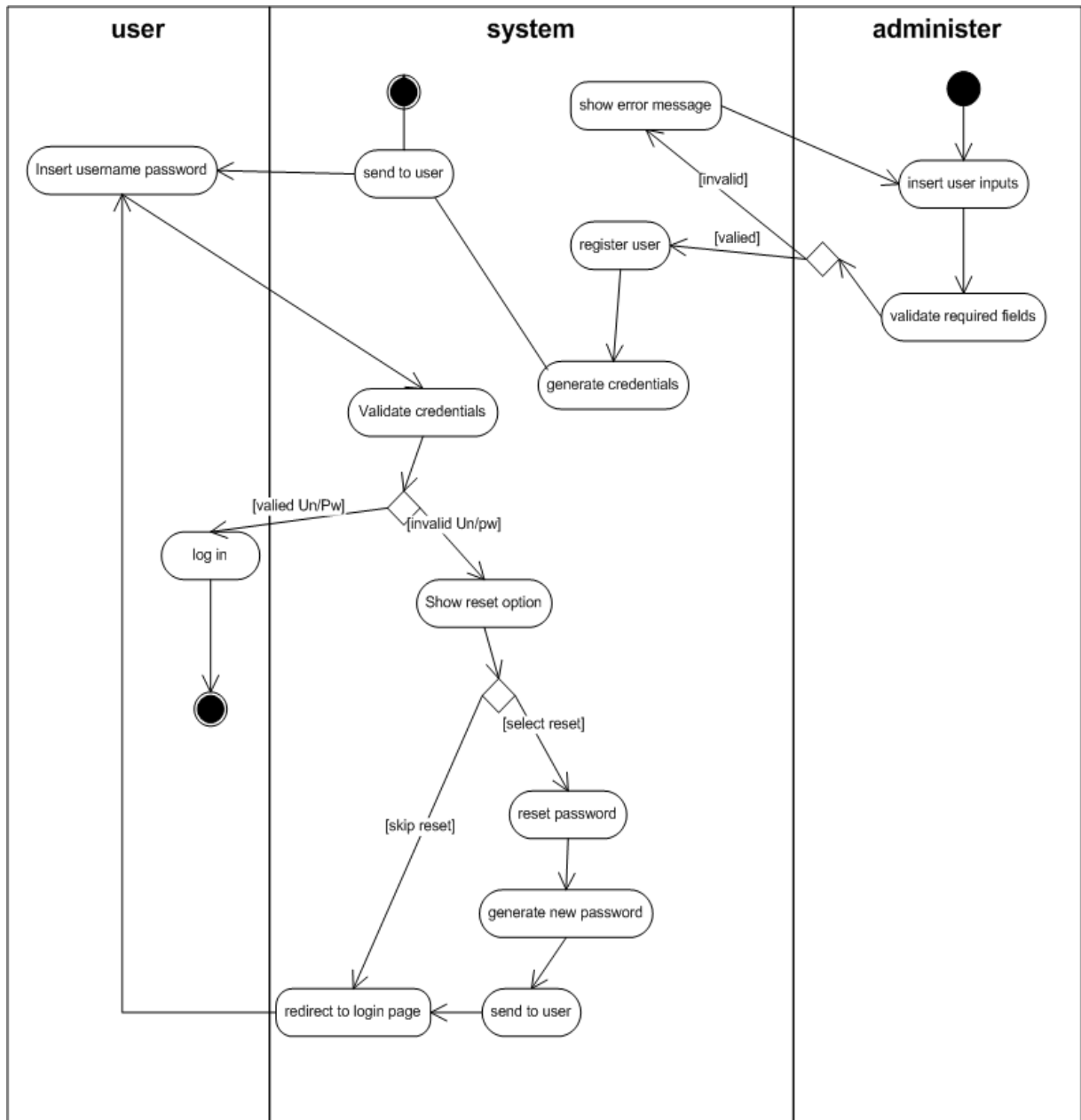


Figure 3.4 - Activity diagram for user login

3.13 State Diagrams

The following diagram (Figure 3.5) shows states for the account object.

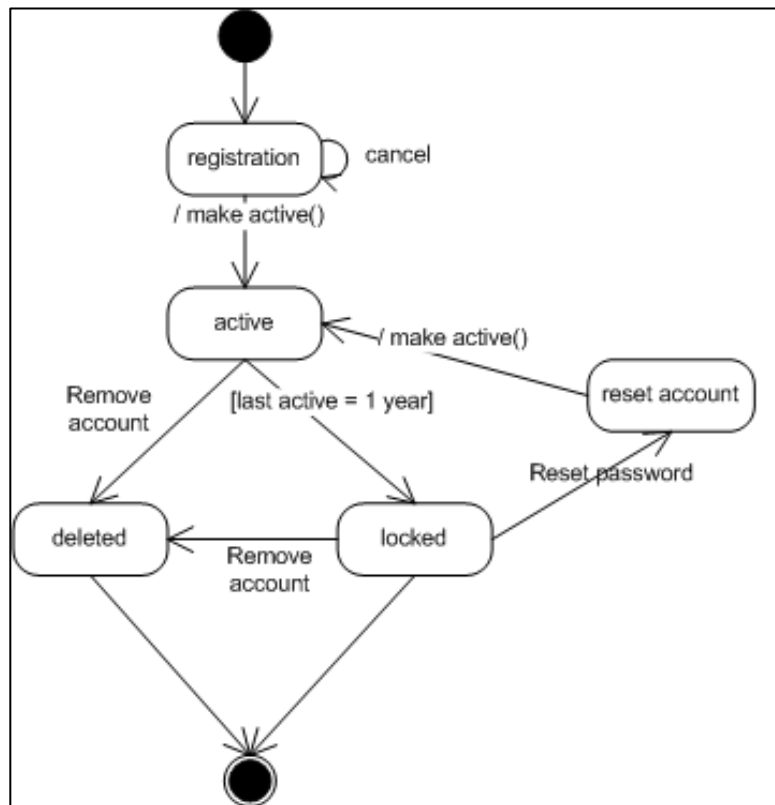


Figure 3.5 - State diagram for the account object

3.14 Design Decisions

Interface design decisions

- Write comments in appropriate places wherever possible.
- Try to modularize much as possible based on the functionality.
- Use of common web service, user control and objects optimally.
- Use standard error messages in the common modules.
- Text box validation messages should be displayed nearby the text box.
- Before deleting a record, confirmation should get using a message from the user.
- To have a better idea of program messages, it should be started from a capital letter and then in simple letters in English.

Database design decisions

Following table describes (Table 3.2) system privileges of each actor. Administrators have most privileges while customers have least access to the database.

Table 3.2 - Database Privilege

User Category/Actor	Self-change password	View All Data	View Own Profile	Update Others Profile	Update Own Profile
Administrator	NO	YES	YES	YES	NO
Employee	NO	NO	YES	NO	Limited
Customer	YES	NO	YES	NO	Limited

3.15 System Interfaces

Few interfaces of the system mentioned below.

Below figure (figure 3.6) shows the home page of the developed system. Customers can register and already registered customers can login to their account through the forms in the home page.

The screenshot shows the home page of the Mirage Colombo Platinum Pier Restaurant online ordering system. The page has a blue and white color scheme. At the top, there is a navigation menu with links for Home, Food Zone, Gallery, My Account, and Contact Us. Below the menu is a banner image of a woman smiling at a restaurant table with wine and food. The banner includes the restaurant's logo and name: "MIRAGE COLOMBO PLATINUM PIER RESTAURANT".

Below the banner, there is a welcome message: "WELCOME TO MIRAGE COLOMBO - PLATINUM PIER RESTAURANT ONLINE ORDERING SYSTEM!". The message encourages users to order food from the Food Zone and register an account for fast ordering and delivery.

The page features two main forms:

- ACCESS YOUR ACCOUNT:** This form includes fields for Email and Password, both marked as required. There is a "Remember me" checkbox and a "Forgot password?" link. Buttons for "Clear Fields" and "Login" are at the bottom.
- REGISTER FOR AN ACCOUNT:** This form includes fields for First Name, Last Name, Email, Password, Confirm Password, Security Question, and Security Answer. All fields are marked as required. The Security Question field has a dropdown menu. Buttons for "Clear Fields" and "Register" are at the bottom.

Figure 3.6- home page

Figure 3.7 shows the shopping cart of this system. Customer can choose foods and add to card in this page.

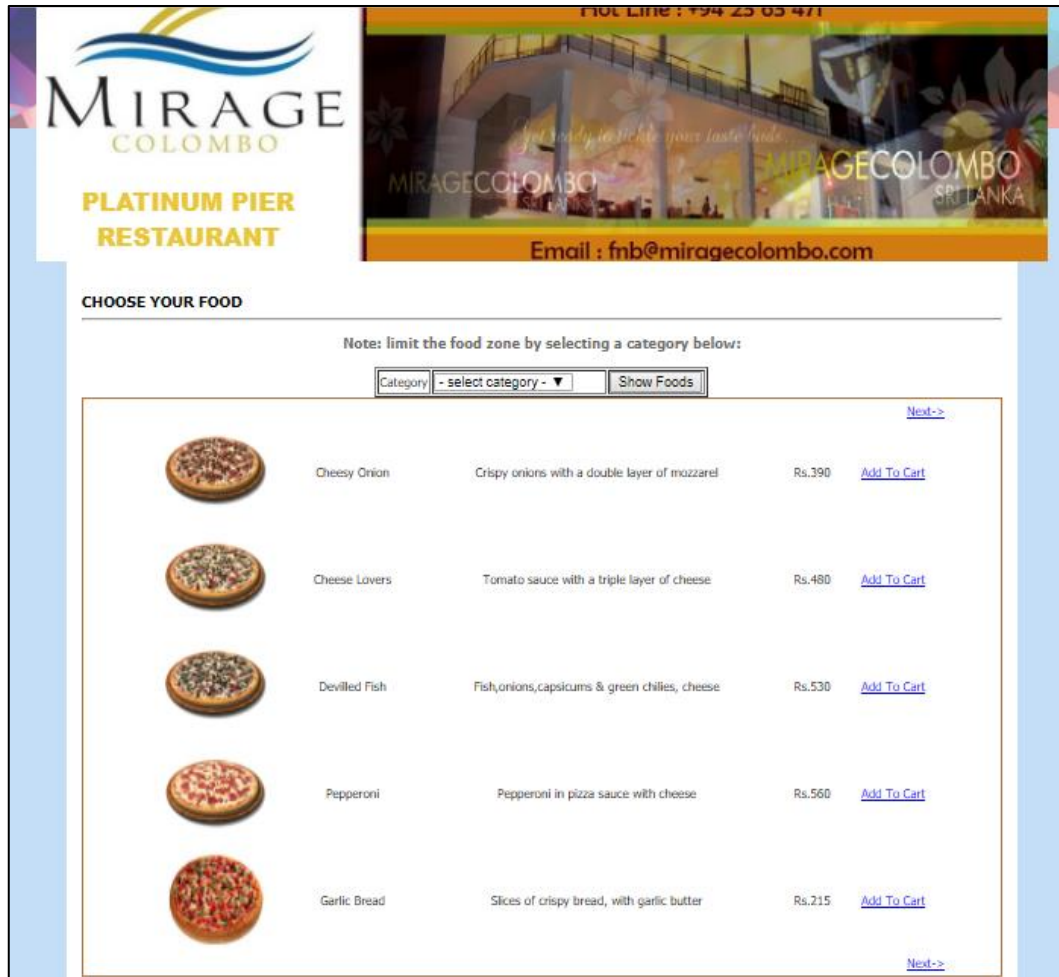


Figure 3.7 – food zone

Below figure (figure 3.8) shows the home page of the admin login.



Figure 3.8 – admin home page

Figure 3.11 shows the report generation interface of the system. Authorized users such as managers who has access to generate reports can login to the system and can generated reports using the below interface.

Orders Management

Home | Categories | Foods | Accounts | > Orders < | Reservations | Specials | Staff | Messages | Options | Content | Customer Details | Logout

View Orders

Start Date:	<input type="text" value="mm/dd/yyyy"/>
End Date:	<input type="text" value="mm/dd/yyyy"/>
<input type="button" value="VIEW"/>	

Generate PDF

Start Date:	<input type="text" value="mm/dd/yyyy"/>
End Date:	<input type="text" value="mm/dd/yyyy"/>
<input type="button" value="Generate PDF"/>	

Orders List

Order ID	Customer Names	Food Name	Food Price	Quantity	Total Cost	Delivery Date	Delivery Address	Mobile No	Actions(s)
Next->									
Next->									

© 2017 PLATINUM PIER RESTAURANT. All Rights Reserved?> Activate

Figure 3.9-report generation

Please refer appendix e for more interfaces of the system.

Chapter 4: Implementation

This chapter describes the development environment of the designed system and how system implemented, language that used for development, the standards followed and development tools that used, also all major codes and development modules and reused components are discussed here. Likewise this chapter explains how user interfaces implemented, the way system validate user inputs and how system generate user reports.

4.1 Software Background

Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. Java language has been used develop the system.

Java server pages & servlets

Java Server Pages / Servlets were used as development languages. Servlets used as controller model and jsp used as view model. And also JSTL 1.1 (JSP Standard Tag Library) and EL (Expression Language) tags were used to develop the system.

MySql

It is free and most popular open source database software over the world, it was developed and maintained by Oracle Corporation. MySql works in client/server or embedded systems. Hence we deiced to use MySql to create the database of the system.

Jquery

jQuery is a lightweight and it make easier to use JavaScript on any website. jQuery also simplifies lot of complicated code from JavaScript, like AJAX calls and DOM manipulation. jQuery elements display even when JavaScript is disabled and it is free.

Fusion Chart

Fusion Charts comes with open-source plugins for libraries like jQuery frameworks and also it can be configured with any proگرامing languages like ASP.NET or PHP. Fusion Charts provide wide range of charts like line, column, pie etc or most complex ones like waterfall, gantt, candlestick, zoomline etc. It allows complete flexibility to customize the charts according

user requirements. With Fusion Charts, it is easy to download or export any chart to the format of JPEG, PNG, PDF or SVG. Hence it was decided to use Fusion chart to display reports of the system.

4.1.1 System Implementation

Hardware and software requirements and other technical requirements needed for the implementation of the developed system is explained below.

Hardware Requirements

Table 4.1-Recommended Hardware Requirements

Recommended Hardware Requirements	
Processor	Intel Pentium IV or higher
Memory	8 GB or higher
Hard disk	10 GB free hard disk space or higher
Internet	Dedicated connection to the internet

Software Requirements

Table 4.2-Recommended Software Requirements

Recommended Software Requirements	
Operating system	Windows OS
Java development kit	Java EE Development Kit 1.7 or newer
Eclipse IDE	Eclipse neon
Database	MySQL 5.0 with MySQL workbench 6
Web server	Wamp server 3.0
Reporting	FusionCharts
Web browser	Google Chrome, Mozilla Firefox

Pre installations required

1. Install MySQL Server 5.0 and MySQL Workbench.
2. Install the Java EE Development Kit (JDK) 1.7 or newer version on system.
3. Install Eclipse IDE with Wamp server

Restore application database

1. Run MySQL Workbench.
2. Log in as root.
3. Click data import/ export from menu and select backup file.
4. Locate the backup file and open it.

4.2 Usage of Design Patterns

Main design pattern was used during the system development was Model-View-Controller (mvc) model. It is the most fundamental design pattern that can be used to develop complex web applications that required maintaining several views of same data.

View - Responsible for the displaying data. User interacts with the view we have used jsp pages as view. It sends the user requests to the controller.

Controller - Gets user input from the view, it analyze the request object and process it with help of models. One controller may or may not use one or several models at a given time to process requests.

Model - keep the real business logic and the state, it knows the rules for getting and updating the own state. Model interact with database and other classes to fulfill controller's requests.

4.3 Development Tools

Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse IDE was used to develop the system.

Adobe Dreamweaver

Dreamweaver is a web designing tool it was used to develop systems web interfaces, css styles and java scripts.

Ms Visio

Microsoft Visio was an advanced design tool that used to design UML and Other diagrams. It was used to design use-case, activity, sequence, collaboration diagrams and ER diagram of the application.

Ms Project

It is a project management software that developed by Microsoft Corporation to manage and develop project plans.

FusionCharts

Is an open-source free flash charting component that can be used to manipulate data-driven animated charts. FusionCharts can be used with any web scripting language like PHP, ASP, .NET, JSP, ColdFusion, JavaScript, and Ruby on Rails etc. All systems graphical reports generated via fusion chart.

SQL Developer

It is free and user friendly software to database manipulations. To develop and design database of the system we have used SQL Developer.

4.4 User Interface Design

When designing an interface, the main goal is to produce a user interface which makes easy to understand, efficient and user friendly to interact with the machine. So operator needs to provide minimal input to obtain required output. The system was designed with minimum number of colors range by following design rules. Users make mistakes most of the time, the system should be able to minimize those mistakes and at the same time it should be able to recover from errors.

When system in design phase developer followed most of UI Design Principles to achieve redundant of user confusion about system operations, as a result system should give instant error messages if a user makes any error or insert invalid input.

4.5 Reusable Codes

Programmers have always reused sections of code, templates, functions, and procedures. Code reuse aims to save time and resources and reduce redundancy by taking advantage of assets that have already been created in some form within the software product development process. The key idea in reuse is that parts of a computer program written at one time can be or should be used in the construction of other programs written at a later time.

DatabaseConnection.java located on database package of this system, will set the parameters and create a database connection. Web.xml file includes database url and user name, password and those context parameters used to get a database connection. It is the most reusable code that used in the system.

4.6 Program List

Some programs related to main functions and validations of the system are listed below.

Below figure (figure 4.1) shows the program of config.php. This program used to create the database connection when needed to insert, update, delete or view data from the database.

```
<?php
    define('DB_HOST', 'localhost');
    define('DB_USER', 'root');
    define('DB_PASSWORD', '');
    define('DB_DATABASE', 'pizza_inn');
?>
```

Figure 4.1-config.php

Below figure (figure 4.2) shows the content of locale.php program. This has used to create the template of the website. This includes the data related to the template used to design the common content of web pages.

```
<?php
//Include database connection details
require_once('connection/config.php');
//Connect to Mysql server
$link = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
if(!$link) {
    die('Failed to connect to server ... The MySQL error is ' . mysql_error());
}
//Select database
$db = mysql_select_db(DB_DATABASE);
if(!$db) {
    die("Unable to select database ... The MySQL error is " . mysql_error());
}
?>
<?php
//retrieving all records from content
$content=mysql_query("SELECT * FROM content");
if ($content){
    //fetch content into an array
    $content_active=mysql_fetch_array($content);
    //directly initializing variables with content values based on fields
    $name = $content_active['display_name'];
    $title = $content_active['home_title'];
    $subtitle = $content_active['home_subtitle'];
    $about = $content_active['about_description'];
    $mission = $content_active['about_mission'];
    $vision = $content_active['about_vision'];
    $contacts = $content_active['contacts'];
    $location = $content_active['contact_location'];
    $specials = $content_active['specials_description'];
    $account = $content_active['myaccount_description'];
    $profile = $content_active['myprofile_description'];
    $inbox = $content_active['inbox_description'];
    $tables = $content_active['tables_description'];
    $partyhalls = $content_active['partyhalls_description'];
    $eating = $content_active['eating_description'];
    $billing = $content_active['others_address'];
    $loggedout = $content_active['others_loggedout'];
    $accessdenied = $content_active['others_accessdenied'];
}
else{
    die("Something went wrong while loading content ... the MySQL error is " . mysql_error());
}
}
```

Figure 4.2-locale.php

Below figure (figure 4.3) shows the php program related to registration form of the website.

```

<CAPTION><h2>REGISTER FOR AN ACCOUNT</h2></CAPTION>
<tr>
  <td colspan="2" style="text-align:center;"><font color="#FF0000">* </font>Required fields</td>
</tr>
<tr>
  <th>First Name </th>
  <td><font color="#FF0000">* </font><input name="fname" type="text" class="textfield" id="fname" maxlength="15" placeholder="provide your first name" required/></td>
</tr>
<tr>
  <th>Last Name </th>
  <td><font color="#FF0000">* </font><input name="lname" type="text" class="textfield" id="lname" maxlength="15" placeholder="provide your last name" required/></td>
</tr>
<tr>
  <th width="124">Email</th>
  <td width="168"><font color="#FF0000">* </font><input name="login" type="email" class="textfield" id="login" maxlength="25" placeholder="provide your email" required/></td>
</tr>
<tr>
  <th>Password</th>
  <td><font color="#FF0000">* </font><input name="password" type="password" class="textfield" id="password" maxlength="25" placeholder="provide your password" required/></td>
</tr>
<tr>
  <th>Confirm Password </th>
  <td><font color="#FF0000">* </font><input name="cpassword" type="password" class="textfield" id="cpassword" maxlength="25" placeholder="repeat your password" required/></td>
</tr>
<tr>
  <th>Security Question </th>
  <td><font color="#FF0000">* </font><select name="question" id="question">
    <option value="select">- select question -
    <?php
      //loop through quantities table rows
      while ($row=mysql_fetch_array($questions)){
        echo "<option value=$row[question_id]>$row[question_text]";
      }
    </select></td>
</tr>
<tr>
  <th>Security Answer</th>
  <td><font color="#FF0000">* </font><input name="answer" type="text" class="textfield" id="answer" maxlength="15" placeholder="enter your answer" required/></td>
</tr>
<tr>
  <td colspan="2"><input type="reset" value="Clear Fields"/>
  <input type="submit" name="Submit" value="Register" /></td>
</tr>

```

Figure 4.3-registration form

Figure 4.4 shows the java script code for the validations of customer's login form.

```

//function to handle login-form validation
function loginValidate(loginForm){

var validationVerified=true;
var errorMessage="";

if (loginForm.login.value=="")
{
errorMessage+="Email not filled!\n";
validationVerified=false;
}

if(loginForm.password.value=="")
{
errorMessage+="Password not filled!\n";
validationVerified=false;
}

if (!isValidEmail(loginForm.login.value)) {
errorMessage+="Invalid email address provided!\n";
validationVerified=false;
}

if(!validationVerified)
{
alert(errorMessage);
}

return validationVerified;
}

```

Figure 4.4-customer login form validation

Please refer appendix f for the more programs of the system.

Chapter 5: Evolution and Testing

To evaluate the final system throughout system development life-cycle, better evaluation techniques required. System testing used to check whether system functioning appropriately and to confirm requirements were satisfied to not.

This chapter describes how the system was tested. And the software bugs that found and the solution to the problems identified. Further describes the test plan and test process, also the expected output and behavior from the test plan.

Software testing is a process of executing a program or system for finding errors. It involves some activity that determining system meets its required results. Software bugs will almost always exist in any software application.

Finding the design defects in software is difficult and testing boundaries are not sufficient to guarantee the correctness of the system. This against the quality of software however, to make better efficiency in software certain quality protocol and rules must be laid down.

But testing can be used to improve quality, verification and validation and also for reliability estimation. There are many tests that can be applied to the software like black box testing and white box testing; also the verification and validation are used for software testing.

5.1 Test Plan

Unit testing

Unit testing is a method by which the source code of the different units is tested to determine if they are fit for use. In object-oriented methodology, unit will be a class; it is the smallest testable part of an application. When each sub systems were completed, unit testing done by the developer.

After completion of each module of the system, unit testing was done to each module to identify the bugs before integrate it as whole. Few sample unit tests carried out for the system are mentioned below.

Table 5.1 – unit test for customer log in

Customer log in				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Click submit button without inserting any field.	To validate empty filed in login page	Show error message “User name / password field cannot be empty!”	Pass
02	Insert password value and skip user name value and click submit.	To test login without user name	Show error message “User name / password field cannot be empty!”	Pass
03	Insert user name value and skip password value and click submit.	To test login without password	Show error message “User name / password field cannot be empty!”	Pass
04	Insert correct user name and incorrect password value and click submit.	To test log in with invalid password	Show error message “Invalid User name or password!”	Pass
05	Insert incorrect user name and correct password value and click submit.	To test log-in with invalid user name	Show error message “Invalid User name or password!”	Pass
06	Try to access customer account page without logged in.	To check whether system able to prevent unauthorized access	Show error message “You are not logged in!!!”	Pass
07	Insert correct user name and password.	To check valid login	Redirect to relevant home page	Pass
08	Click log out button after logged in	To check whether successfully logged out or not	Show logout message with hyper-link to go to login page	Pass

Table 5.2 – unit test for add new food category

Add new food category				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Click submit button without entering a category name.	To validate empty filed in category adding form	Show error message “Category name cannot be empty!”	Pass
02	Insert password value and skip user name value and click submit.	To test login without user name	Show error message “User name / password field cannot be empty!”	Pass
03	Insert a duplicate category name and click submit.	To test inserting duplicate values	Show error message “Entered food category is already exists.”	Pass

Black box testing

It is not based on internal code or design those tests are based on the requirement and functionality of the system. After completing the each sub system and after completing the application, black box testing will be used to check whether required goals are met.

Black box testing done for each sub system of the proposed system after completing each. Input the relevant data through the forms in sub system and check whether it provides the expected outputs. And the bugs that were prevailing in some forms of the sub systems were fixed. Black box testing was useful to identify the below issues in the developed system.

- Incorrect or missing functions
- Interface errors
- Errors in data structures
- Behavior or performance errors
- Initialization and termination errors

White box testing

Done by knowing the internal code and business logic of the system. Test is coverage of code segments, paths, conditions. White box testing will use while developing the system and end of each sub system. Finally after completing system, full code walk through will do to find any existing errors.

Using white box testing methods, the following tests were made on the developed system.

- All independent paths within a module have been exercised once. In the developed system, ensuring that case was selected and executed, checked all case structures. The bugs that were prevailing in some parts of the code were fixed.
- All logical decisions were checked for the truth and falsity of the values.

System Integration Testing

The order of progress in testing which several system elements are developing together and tested. This testing proceeds until the entire system has been developed.

Integration testing makes sure that the interfaces of the each component of the system are functioning properly. This testing will do progressively through the development, until all the sub systems are properly integrated until the software works as a system.

Below mentioned scenarios were checked using the system integration testing.

- To check whether the modules developed individually when combined are according to standards and gives the expected results.
- When modules are combined, sometimes the data travelling between modules has many errors which may not give the expected results. So integration testing is performed to find the defects or bugs in all the interfaces.

System Testing

The methods of testing an integrated hardware and software system to make sure that the system meets its planned requirements.

Various complex test cases were used to test the developed system according to the business process requirements which are collected from the user. Also the system put in to use by few staff members at the restaurant for a selected period and monitored the system. Few test customer logins and staff logins created and they were asked to enter the orders of the customers who visited the restaurant, to the new system and to handle the order processing also using the new system. This method was very useful to do the system testing to identify the bugs and errors of the system.

Selenium IDE

Selenium IDE is a free test automation tool that used to develop custom test cases. It is implemented as a Firefox extension, and allows to record, edit, and debug tests. It is very easy to use and has the capability to convert the test to different programming languages such as html, java also provide Logging capabilities. It make easier to do smoke test while developing web based software.

5.2 Test Plan of the System

Test cases were designed to make sure all aspects of the system functioning correctly. Several test cases were designed to check the system. System was tested by sub module vice because of the complexity of test module.

Table 5.3 – Test Plan

Sub module	Events
Administer	Log in
	Manage customer messages
	Manage devices
	Manage transactions
	Generate Reports
	View customer orders
Super admin	Manage Administrators
Customer	Create orders
	Pay bill
	View orders
	Contact admin

5.3 Test Cases

Following table shows test case for Administrator log in.

Table 5.4 - Test case for administrator log in

Administrator log in				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Click submit button without inserting any field	To validate empty filed in login page	Show error message “User name / password field cannot be empty!”	Pass
02	Try to insert special characters in user name field	To check if system able to face sql injection	Show error message “User name has invalid characters!”	Pass
03	Insert password value and skip user name value and click submit.	To test login without user name	Show error message “User name / password field cannot be empty!”	Pass
04	Insert user name value and skip password value and click submit.	To test login without password	Show error message “User name / password field cannot be empty!”	Pass
05	Insert correct user name and incorrect	To test log in with invalid password	Show error message “Invalid User name or password!”	Pass

	password value and click submit.			
06	Insert incorrect user name and correct password value and click submit.	To test log-in with invalid user name	Show error message "Invalid User name or password!"	Pass
07	Try to access administrator home page without logged in.	To check whether system able to prevent unauthorized access	Show error message "You are not logged in!!!"	Pass
08	Insert correct user name and password	To check valid login	Redirect to relevant home page	Pass
09	Click log out button after logged in	To check whether successfully logged out or not	Show logout message with hyper-link to go to login page and after 5 second redirect to login page	Pass

Following table shows test case for Customer create orders.

Table 5.5 - test case for Customer create orders

Create orders - customer				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Click submit without inserting required values	To check whether all filed are needed to create order.	Display error message "required fields must be entered" and unable to create order.	Pass
02	Insert food item and do not insert the other fields.	To check whether all filed are needed to create order.	Display error message "required fields must be entered" and unable to create order.	Pass
03	Insert all the fields except the quantity.	To check whether quantity need to enter or not.	Display error message "Quantity must be entered" and unable to create order.	Pass
04	Insert all the fields except the payment method.	To check whether payment method need to enter or not.	Display error message "Payment method must be entered" and unable to create order.	Pass
06	Insert all the data correctly & click submit.	To check whether order successfully created or not.	Redirect to relevant payment form.	Pass

Refer appendix d for more test cases.

5.4 User Evaluation

User evaluation report provides an overview of how user is satisfied with the system. It includes both functional and non-functional user requirements and it is important to get user feedback in order to successfully complete the project.

Some statistics related to user evaluation report is explained below.

Below figure (figure 5.1) shows the satisfaction level of the user according to the user evaluation report. 85% of the users very much satisfied about the developed system.

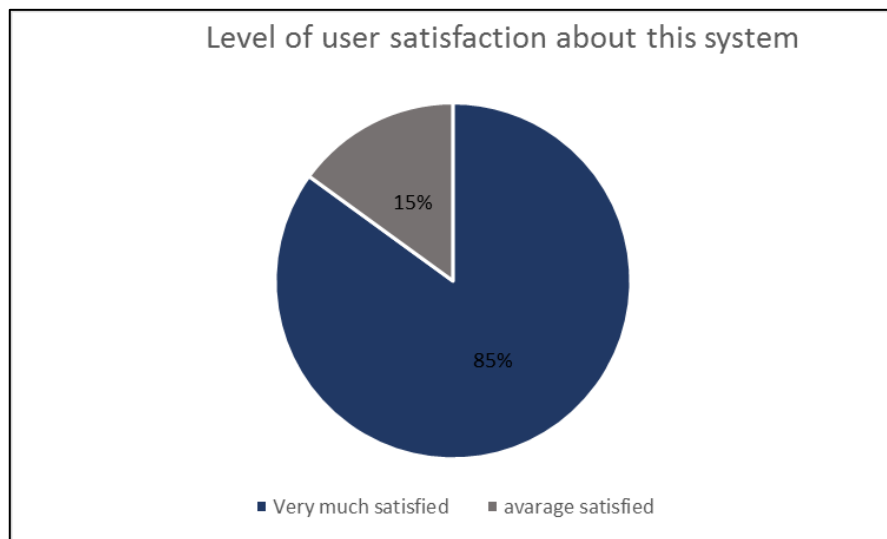


Figure 5.1-stat of user satisfaction level

According to the user evaluation, level of overall system performance and the level of fulfillment of the stated objectives shows in the Figure 5.2.

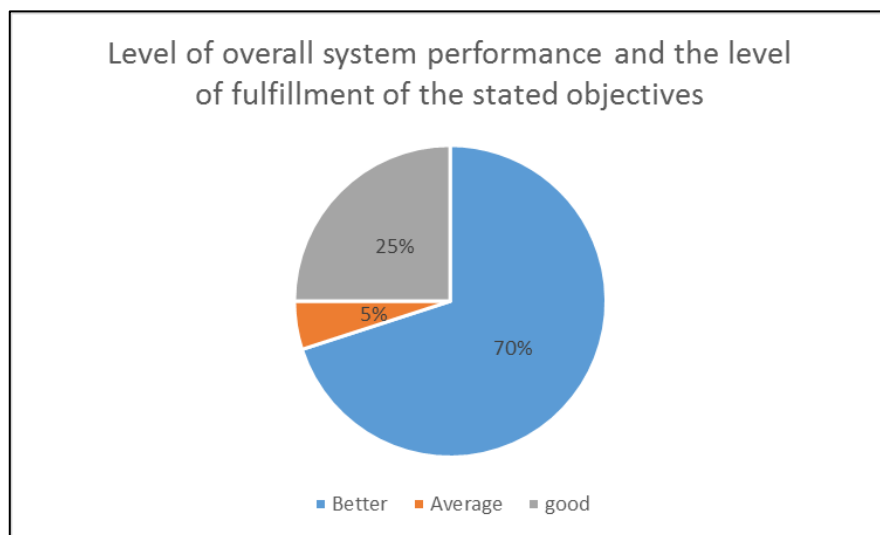


Figure 5.2 – stats of user evaluation of system performance

Below graph (figure 5.3) shows the level of customer satisfaction with related to the user friendliness of the developed system. Most of the users are highly satisfied with the developed system according to the user evaluation report.

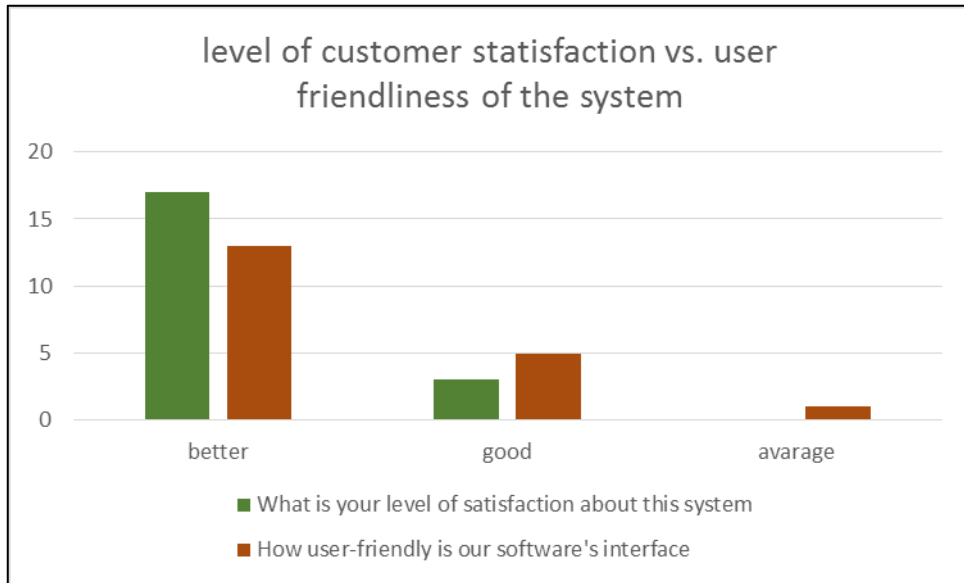


Figure 5.3- customer satisfaction vs. user friendliness

Refer Appendix B for User Evaluation Form.

Chapter 6: Conclusion and Future Work

The requirement was set out to explore the concept of make efficient online restaurant management and food ordering system for Mirage Colombo Hotel. Use of the system is cost effective and significant amounts of time and money can be saved, by implementing this automated system, it will reduce costs and improve transparency of the process so any failure or inconvenience on the consumer side can be instantly detected and rectified.

This chapter describes project assessment, problems that found while developing system, lessons learned and possible future enhancement for the developed system.

6.1 Project Assessment

To system to be a success, it should meet all critical requirements of the users and fulfill all the objectives that discussed in Chapter 1. After carefully evaluating problem domain, I have identified requirement of web based food ordering system. The project started with gathering and understanding the requirements of the customers and admin users of the system.

After fully understanding the logic behind the food ordering and restaurant management process, which gathered from the employees working in the company and analyzing existing system documents, several alternative solutions were considered while implementing the system. Finally Java and JSP/Servlets had been selected as the most suitable solution to complete the project.

The main objective was to reduce the queues in front of the cashiers and to gain more business to the company by developing advanced system that enables customer to order at home just by logging to a web site.

Complete a literature survey and technical background analysis were done to have extended idea about current technological concepts that available in the world. In requirement gathering phase, we identified critical functional and non-functional requirements of the systems further more hardware and other special requirements of the system.

System designed by following object-oriented methods, numerous entity relationship diagrams and several other UML diagrams were designed to get a better understanding of the system process. Many software tools (mentioned in implementation chapter) were used to develop final software system.

System developed by using advanced development tools and technologies to make more user friendly to the end users. System reports designed to make optimal solutions to given scenarios and at the same time system designed to generate reports faster, in numerically or graphically to the final users. Many test cases and test automation tools used while implementing the system to make minimize systematic errors and bugs.

In the process of implementing the system, the author acquired a thorough knowledge and experience in the software development life cycle and self-confidence to complete the project successfully on time.

6.2 Problems Encountered

- Time was the main concern encountered while developing a system. Some module took more time than previously planned. Hence WBS time frame rescheduled to complete system on time.
- In analysing and requirement gathering phase it was difficult to identify correct and most suitable business logic to the system. So there was a need of prototype.
- When designing prototypes author had to identify and analyse the current system functionality repeatedly to have a better understanding of the system, furthermore had a difficulty in selecting a bank payment gateway hence selected PayPal as payment gateway.
- In the analysis phase it was decided to remove some functions, at the same time some new critical functions were added to the system.
- At the beginning of the project the knowledge which I had about online food ordering system was limited. However, in the process of the development of the system knowledge about the field had been substantially enhanced.
- Interface designing was challenging, it had been modified several times to match user requirements.
- System was developed following standard interface designing methods, however, there was minor modifications in some pages because of some methods were not suitable for required interfaces.
- Due to lack of knowledge in JSP, servlet and other technologies used, it took more time than predicted when developing the critical functionality of the system.

6.3 Lessons Learned

In order to finish the project, according to the guidelines, the author had to self-study to learn about the software development life cycle.

The author acquired good knowledge and experience in implementing software system, programming languages, development methods and theories after using them for system development furthermore the author gained an excellent knowledge of tools that used to develop the system. Most of the development was done by trial and error.

Software testing methodologies and project management methods were able to apply that learned from previous years; the author was able to gain a better understanding and practical experience from them.

Also author learned the process of proper documentation methods while writing this dissertation. The author had to put extra effort and time during documentation. It improved authors writing skills and learning methods.

While developing the system author had to select suitable software, hardware and development environments make the system at the same time, the author had to learn the Java, Eclipse IDE, fusion chart, MySQL and may other technologies.

6.4 Achievements

The main objective of this project was to automate the manual work of the restaurant of Mirage Colombo. By implementing the developed system at platinum pier restaurant, able to automate the whole restaurant management process and to provide online food ordering facility to customers. This reduces the long queues at the counter and increase the sales. Also restaurant staff is able to provide a better service than before for those who visit the restaurant. This increases the customer satisfaction of those who visit the restaurant. Since the customers can order food from anywhere at any time and get it deliver to the place they want, customer satisfaction has more increased.

Since the customers can provide their views regarding the online ordering and restaurant via the new system, customer relationship managements is also become more easy and accurate. And also it has increased the customer relationship.

Major achievement of this project is, thought this online food ordering and restaurant management system developed to restaurant at mirage Colombo hotel, this is developed to use in any restaurant. This can be implement in any restaurant after customizing the system according to the needs of the restaurant. This system is developed with the flexibility to customize according to any restaurant. Even an admin user can customize the template of this system also using the options given in the admin panel.

The system has reduce and replace the human manpower task, reduce the time consume for each transaction and generate report for further management purpose by fully utilizing the system.

The developed system helped to improve the productivity of the restaurant and thus directly did an impact to the profitability of the restaurant. Furthermore, it also helped restaurant to reduce the cost of operation in term of manpower, because the system have already facilitate majority of the business process by using the system. Therefore, it is believed that the system can lead the restaurant's business grow from time to time.

6.5 Future Developments

- The system can develop using advanced frameworks such as hibernate or struts then code segments can be reduced and improve more stability of the system.
- It is recommended to buy an SMS gateway to send receive SMS messages.
- Current system mange payments by using PayPal payment gateway, it was selected as payment gateway represents only so recommend to buy payment gateway access from Sri Lankan banks and configure system, so local customer to make the payments.
- Some old browsers do not support css3 functions; however, most of current browsers now support new css3 functions. It will be possible to improve user interfaces of the system by applying advanced CSS3 options in the future.
- Today the trend is mobile applications and people are more tend to use mobile apps than logging into a website using a laptop or PC since most of the people are bringing smart phone with them. Hence recommended to develop a mobile app for this in future.

References

- [1] Colombo Hotels Sri Lanka, Mirage Colombo is a 4 star luxury hotel in Colombo Sri Lanka, Miragecolombo.com, [Online]. Available: <http://www.miragecolombo.com>.
- [2] Ashutosh, B., Niranjana, J., Apurva, J., Prachi, O. and Lahane, S. (2013). Digital Ordering System for Restaurant Using Android. [Online] www.ijsrp.org. Available at: <http://www.ijsrp.org/research-paper-0413/ijsrp-p1605.pdf>. [Accessed 16 Nov. 2016].
- [3] McDelivery™ Sri Lanka, Mcdelivery.lk, [Online]. Available: <http://www.mcdelivery.lk>.
- [4] Rashid, M., Izzuddin, T., Abas, N., Hasim, N., Azis, F. and Aras, M. (2013). Control of Automatic Food Drive-Through System using Programmable Logic Controller (PLC). [Online] www.sersc.org. Available at: http://www.sersc.org/journals/IJUNESST/vol6_no4/4.pdf [Accessed 05 Jan. 2017].
- [5] Pizza Hut, Pizza Hut, [online]. Available: <http://www.pizzahut.lk>.
- [6] Khairunnisa, K., Ayob, J., Mohd. Helmy, A., Erdi Ayob, M., Izwan Ayob, M. and Afif Ayob, M. (2009). The Application of Wireless Food Ordering System. [Online] Available at: http://eprints.uthm.edu.my/5726/1/Wireless_Food_Ordering_System.PDF [Accessed 10 Dec. 2016].

Appendix A – System Documentation

System Documentation

Title : Online Restaurant Management & food ordering system
Version : 1.0
System : Web application

This section of the document provides necessary details on technical requirements of the system configuration, installation and other technical information to system administrators, developers and people who like to improve the system or add new features to the system. Please refer recommended hardware (Table A.1) and software (Table A.2) requirements for the system.

Hardware Requirements

Recommended Hardware Requirements	
Processor	Intel Pentium IV or higher
Memory	8 GB or higher
Hard disk	10 GB free hard disk space or higher
Internet	Dedicated connection to the internet

Table A. 1 Recommended Hardware Requirements

Software Requirements

Recommended Software Requirements	
Operating system	Windows OS
Java development kit	Java EE Development Kit 1.7 or newer
Eclipse IDE	Eclipse neon
Database	MySQL 5.0 with MySQL workbench 6
Web server	Wamp server 3.0
Reporting	FusionCharts
Web browser	Google Chrome, Mozilla Firefox

Table A. 2 Recommended Software Requirements

Pre installations required

4. Install MySQL Server 5.0 and MySQL Workbench.
5. Install the Java EE Development Kit (JDK) 1.7 or newer version on system.
6. Install Eclipse IDE with Wamp server

Restore application database

5. Run MySQL Workbench.
6. Log in as root.

7. Click data import/ export from menu and select backup file.
8. Locate the backup file and open it.

Appendix B - User Evaluation Form

User Evaluation Form for Online Restaurant Management & Food Ordering System

(These details are required for communication purposes only and will not be disclosed)

1. Personal details

Name

Designation

Email

(Please mark the relevant box)

2. What is your level of computer and internet literacy?

Average use **Basic use** **Advanced user**

3. How often you do you order food online?

Daily **Weekly** **Monthly** **Rarely**

4. How user-friendly is our software's interface?

Extremely user-friendly
 Very user-friendly
 Moderately user friendly
 Slightly user-friendly

5. Do you find the system demonstration performed was adequate for the evaluation?

Yes **No** **somewhat**

6. Level of overall system performance and the level of fulfillment of the stated objectives.

Better **Average** **Good** **Unsatisfied**

7. Do you prefer to use online payments option to make payments?

Yes **No**

8. Are icons that are used to assist navigation?

Yes **No**

9. Do you prefer in sharing your knowledge and expertise with others?

Yes **No**

10. What is your level of satisfaction about this system?

Very much satisfied **Average Satisfied** **Unsatisfied**

Appendix C - Use case narratives

Following table (Table C.1) describes use case narrative for customer log in.

ID: U01	Actors: customer Title: Login
Description:	Relevant customer comes to login page Logged in using his/her user name and password
Preconditions:	User has already created an account Customer has a valid user name and password User is in login page
Main Flows:	Enter a valid user name Enter valid password User logged into the system
Post Conditions:	Login successfully and Customer redirect to his/her home page
Alternative paths:	User enters invalid user name, then an error message will be displayed User enters an invalid password, then an error message will be displayed User enters only user name, then an error message will be displayed User enters only password, then an error message will be displayed Customer clicks rest password option Customer should re-enter log in details or terminate the process

Table C: 1 Use case narrative for customer log in

Following table (Table C.2) describes use case narrative for how customer create his/her accounts.

ID: U02	Actors: Customer Title: Create customer account
Description:	Describes how Customer create his/her profile
Preconditions:	User visit the relevant website User is in create account page
Main Flows:	<i>Create account</i> Customer should fill all required fields with valid values Email should have “@” sign and “.” to be a valid email Mobile number should contain ten digits Passwords should match with confirm password field Success message displayed
Post Conditions:	Account details will update in the accounts info table Customer can login later using the relevant user name & password

Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then display error messages
	Click other link when creating user profile then discard changes <ul style="list-style-type: none"> - If click yes then deletes the record. - If click cancel then cancel the operation.

Table C: 2 Use case narrative for create customer account

Following table (Table C.3) describes use case narrative for place order.

ID: U03	Actors: Customer Title: Place order
Description:	Describes how customer create orders
Preconditions:	User logged into the system or visit as a guest user
	User is in order items page
	Registered users have a valid email updated in his/her account
	User connected to internet
Main Flows:	User select order type
	User select items
	User enter quantity
	Click add to cart
	Success message displayed
Post Conditions:	Bill generate for the relevant order
	Order details updated in the order table
Alternative paths:	User navigate to other page while creating the account then discard changes
	User keeps a blanks to required fields then error message will display
	User enter an invalid e mail then display error message
	User enters password without the requested conditions then display error message

Table C: 3 Use case narrative for place order

Following table (Table C.4) describes use case narrative for view reports by Employee.

ID: U04	Actors: Employee Title: Generate reports
Description:	Describes how employee generate system related reports
Preconditions:	User logged into the system
	User is in generate report page
	There should be at least one data record in database
	JavaScript must be enabled
Main Flows:	User enters relevant details required for the report
	Click view button
	Display report.
Post Conditions:	User can click export option as pdf or jpg image format

	Input values can be added again to generate new report
Alternative paths:	User navigate to other page when selected only one input value then discard changes
	User keeps a blank to the input fields display error message
	If there any error in generating report display error message

Table C: 4 Use case narrative for generate reports

Following table (Table C.5) describes use case narrative for customer payments.

ID: U05	Actors: Customer Title: Pay bill
Description:	Describes how Customer make an online payment
Preconditions:	User logged into the system or visit as a guest user
	User is in pay online page
	Customer must have PayPal account
Main Flows:	User view his account due value
	User enters payment field value
	Click pay button
	Redirect to PayPal payment gateway page
	Log in to PayPal account
	Click Pay Now button
	Display payment information
	Click redirect back to option
	Display payment success message with local payment serials number
Post Conditions:	Reduce payment from customer account due balance and display
	Payment records display later on customer reports
Alternative paths:	User enters payment field value as negative value and click pay button then display error message
	User enters payment field value as character value and click pay button then display error message
	User enters payment field value as blank value and click pay button then display error message
	Click cancel in payment gateway then redirect back and display payment cancel information with local serial number

Table C: 5 Use case narrative for customer pay bill

Following table (Table C.6) describes use case narrative for manage user accounts.

ID: U06	Actors: Administrator Title: Manage user accounts
Description:	Describes how administrator manages the user accounts in the system
Preconditions:	Administrator is in Manage user accounts page

	If the administrator adding a new employee user name should be a new user name
	Must have a valid email
	User type already added to the system
Main Flows:	<i>Add new user</i>
	User types an unique user name
	User types a valid email
	Password should match with confirm password field
	Administrator should fill all required fields with valid values
	Administrator should select user type
	Success message displayed
	<i>Update user</i>
	Administrator enters user name in search user field
	Select relevant user from search results and click update button
	Administrator unable to update User id
	Administrator should fill all required fields with valid values
	Administrator should select user type
	Email should have “@” sign and “.” to be a valid email
	Password should match with confirm password field
	Status can be updated
	Success message displayed
	<i>Delete User</i>
	Administrator enters user name in search user field
	Select relevant user from search results and click delete button
	Display confirmation messages
	Click yes
	Administrator return to manage user accounts page
	display success message
Post Conditions:	Records added to login table
	Password encrypted
	User able to login to the system and make changes
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as input then display error messages
	Try to search empty value then display an error message
	Click other link when updating user profile then discard changes
	Click delete button and show confirm message. <ul style="list-style-type: none"> – If click yes then deletes the record. – If click cancel then cancel the operation.

Table C: 6 Use case narrative for manage user accounts

Appendix D – Test cases

Following table (Table D.1) describes test case for payments by customer.

Make payment				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Click pay button without inserting any values	To check whether blank fields validated or not	Display error message “please insert the payment value.”	Pass
02	Click pay button after inserting special characters only	Test sql injection and to confirm system accept only positive numerical values	Display error message “Please enter numerical value” and unable to make the payment	Pass
03	Click pay button after inserting special characters with numerical value	Test sql injection and to confirm system accept only positive numerical values	Display error message “Please enter numerical value” and unable to make the payment	Pass
04	Click pay button after inserting alphabetic characters with numerical value	Test sql injection and to confirm system accept only positive numerical values	Display error message “Please enter numerical value” and unable to make the payment	Pass
05	Click pay button after inserting alphabetic characters only	to confirm system accept only positive numerical values	Display error message “Please enter numerical value” and unable to make the payment	Pass
06	Click pay button after inserting numerical value	To test payment page redirect to merchant or not	Redirect to PayPal payment gateway with current payment information to make the payment	Pass
07	Type invalid details to the merchant portal	To confirm payment information	Unable to access user account	Pass
08	Click cancel in merchant portal	To check payment status now and redirection page	Redirect to payment page display payment failed message with locally generated serial number	Pass
09	Click pay now button and make the payment in merchant portal	To confirm payment	Make the payment	Pass
10	Click return URL in merchant portal	To confirm redirection page and success message	Redirect to relevant page and show payment success message with payment information.	Pass

Table D: 1 Test case for customer payments

Following table (Table D.2) describes test case for create account by customer.

Create account – customer				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Insert name as numerical value	To avoid numerical values in name field	Display error message “Please enter characters only”	Pass
02	Insert name with special characters	To avoid special character values in name field	Display error message “Please enter characters only”	Pass
03	Remove name field value and try to submit	To check form submission without name field accepted or not	Display error message “Name field cannot be empty”	Pass
04	Add name value with all other required fields filled with valid values	To check submission with name field accepted or not	Successfully updated message displayed	Pass
05	Insert mobile no as numerical value	To check mobile no field accepted in what type of characters	No error message displayed, number inserted to database	Pass
06	Insert mobile no with special characters	To check mobile no field accepted in what type of characters	No error message displayed, number inserted to database	Pass
07	Remove mobile no field value and try to submit	To check submission without mobile no field accepted or not	Display error message “Mobile no field cannot be empty”	Pass
08	Add new mobile no value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
09	Insert address as numerical value	To check address field accepted in what type of characters	Accepted new value and display successful message	Pass
10	Insert address with special characters	To check address field accepted in what type of characters	Accepted new value and display successful message	Pass
11	Remove address field value and try to submit	To check submission without address field accepted or not	Display error message “Address field cannot be empty”	Pass
12	Add new address value with all other required fields filled with valid values	To validate fields	Accepted address value and display successful message	Pass

13	Insert invalid format email and fill all other required fields and then click submit	To check email format accepted in email field	Display error message “please enter valid email”	Pass
14	Remove email value and try to submit	To check required filed validation	Display error message “email cannot be empty”	Pass
15	Add new email value with all other required fields filled with valid values	To enter successful record	Record added to the database and display success message	Pass
16	Insert password only using numerical value	To check password field accepted in what type of characters	New value accepted and move on to confirm password field	Pass
17	Insert password with special characters	To check password field accepted in what type of characters	New value accepted and move on to confirm password field	Pass
18	Remove password field value and try to submit	To check password always needed or not	Display error message “password cannot be empty”	Pass
19	Try to add different values to password and confirm password fields	To check both filed matching	Display error message “passwords are not the same”	Pass
20	Try to update password with matching values in password and confirm password fields.	To enter successful record	Record added to the database and display success message	Pass
21	Try to submit without filling all required field values.	To check required filed validation	Display error message in all required fields	Pass
22	Leave a blank in Gender field and try to submit	To check gender field accepted in what type of characters	Successfully update profile	Pass
23	Select the gender from the drop down menu with all other required fields filled with valid values	To enter successful record	Record added to the database and display success message	Pass

Table D: 2 Test case for customer create account

Following table (Table D.3) describes test case for Manage user accounts by administrator.

Manage user accounts- administrator				
No	Test case data	Purpose	Output	Status Pass / Fail
01	Try to insert already existed user id as user name	Prevent unauthorized access and updating same user id again	Unable to update	Pass

02	Insert user name with special characters	To check if system able to face sql injection	Show error message "User name has invalid characters!"	Pass
03	Try to insert blank user name	To validate fields	Display error message "Name field cannot be empty"	
04	Insert name as numerical value	To avoid numerical values in name field	Display error message "Please enter characters only"	Pass
05	Insert name with special characters	To avoid special character values in name field	Display error message "Please enter characters only"	Pass
06	Remove name field value and try to submit	To validate fields	Display error message "Name field cannot be empty"	Pass
07	Add new name value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
08	Insert contact no as numerical value	To check contact no field accepted in what type of characters	No error message displayed, number inserted to database	Pass
09	Insert contact no with special characters	To check contact no field accepted in what type of characters	No error message displayed, number inserted to database	Pass
10	Remove contact no field value and try to submit	To check contact no field accepted in what type of characters	No error message displayed, number removed from database	Pass
11	Add new contact no value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
12	Insert address as numerical value	To check address field accepted in what type of characters	Accepted new value and display successful message	Pass
13	Insert address with special characters	To check address field accepted in what type of characters	Accepted new value and display successful message	Pass
14	Add new address value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
15	Insert mobile as numerical value	To check mobile field accepted numerical characters	Accepted new value and display successful message	Pass
16	Insert mobile with special characters	To check special characters accepted in mobile field	Display error message "mobile number should have numerical values only"	Pass

17	Remove mobile field value and try to submit	To check required filed validation	Display error message “mobile number cannot be empty”	Pass
18	Add new mobile value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
19	Insert invalid format email and fill all other required fields and then click submit	To check email format accepted in email field	Display error message “please enter valid email”	Pass
20	Remove email value and try to submit	To check required filed validation	Display error message “ email cannot be empty”	Pass
21	Add new email value with all other required fields filled with valid values	To validate fields	Records added to the database and display successful message	Pass
22	Insert password only using numerical value	To check password field accepted in what type of characters	New value accepted and move on to confirm password field	Pass
23	Insert password with special characters	To check password field accepted in what type of characters	New value accepted and move on to confirm password field	Pass
24	Remove password field value and try to submit	To check password always needed or not when user update his profile every time	Password not changed	Pass
25	Add new password value without submitting confirm password value	To check password change when he update new password	Password updated	Pass
26	Try to add different values to password and confirm password fields	To check both filed matching	Display error message “passwords are not the same”	Pass
27	Try to update password with matching values in password and confirm password fields.	To validate fields	Records added to the database and display successful message	Pass
28	Try to submit without filling all required field values.	To check required filed validation	Display error message in all required fields	Pass
29	Leave a blank in comments field and try to submit	To check comments field accepted in what type of characters	Successfully update profile	Pass
30	Insert comments only using numerical value	To check comments field accepted in what type of characters	Successfully update profile	Pass

31	Insert comments only using special characters	To check comments field accepted in what type of characters	Successfully update profile	Pass
32	Add new comments value with all other required fields filled with valid values	To enter successful record	Record added to the database and display success message	Pass
33	Try to search inserting blank key	To check whether user able to get a result	No search results displayed	Pass
34	Type character in search field	To check search option	Display search result when user types a character	Pass
35	Click update button on search result	To check update URL	Redirect to update page	Pass
36	Click delete button on search result	To check functionality of delete button	Display confirmation button	Pass
37	Click cancel in delete button event	To check functionality of delete button	Cancel deletion	Pass
38	Click Ok in delete button event	To check functionality of delete button	Delete record and display successful message	Pass
39	Add new password value without submitting confirm password value in update page	To check password change when he update new password	Display error message "passwords are not the same"	Pass
40	Try to add different values to password and confirm password fields in update page	To check both filed matching	Display error message "passwords are not the same"	Pass
41	Try to update password with matching values in password and confirm password fields in update page	To enter successful record	Record added to the database and display success message	Pass
42	Select status as active	To check status update in user profile	Activate account	Pass
45	Select status as inactive	To check status update in user profile	Deactivate account	Pass

Table D: 3 Test case for Manage user accounts

Appendix E – System interfaces

Below figure (figure E.1) shows the shopping cart of the customer.



Figure E.1 – shopping cart

Figure E.2 shows the order history of the customer. Customer can view this after login to the system only.

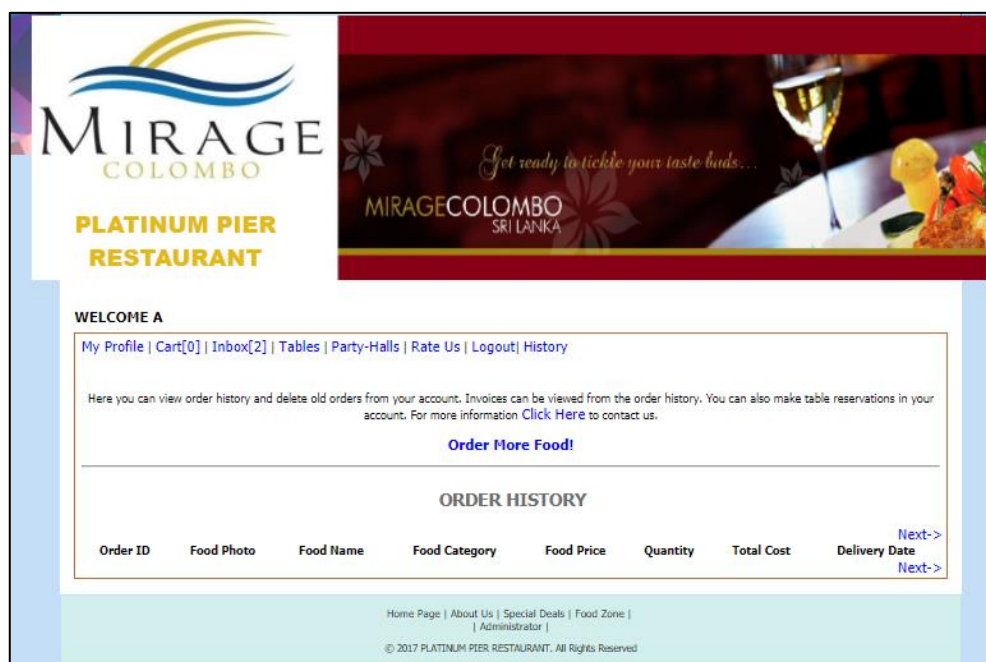


Figure E.2-order history

Figure E.3 shows the categories management interface of the system.

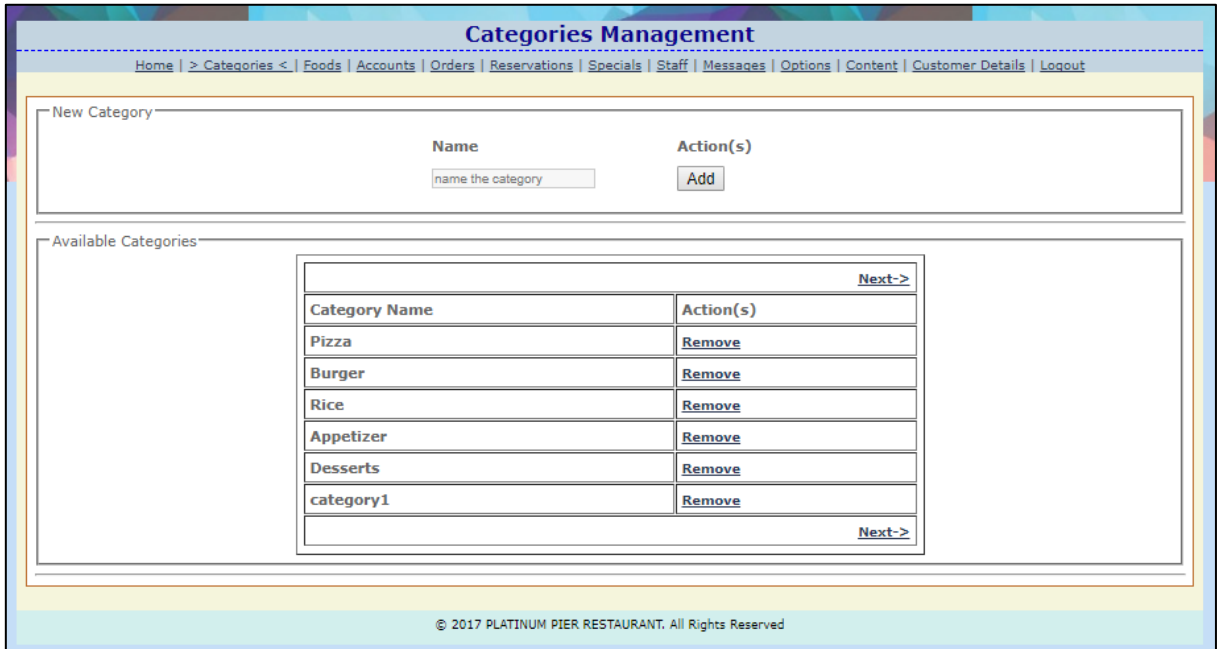


Figure E.3-category update

Below figure (figure E.4) shows the food items management interface of the system.

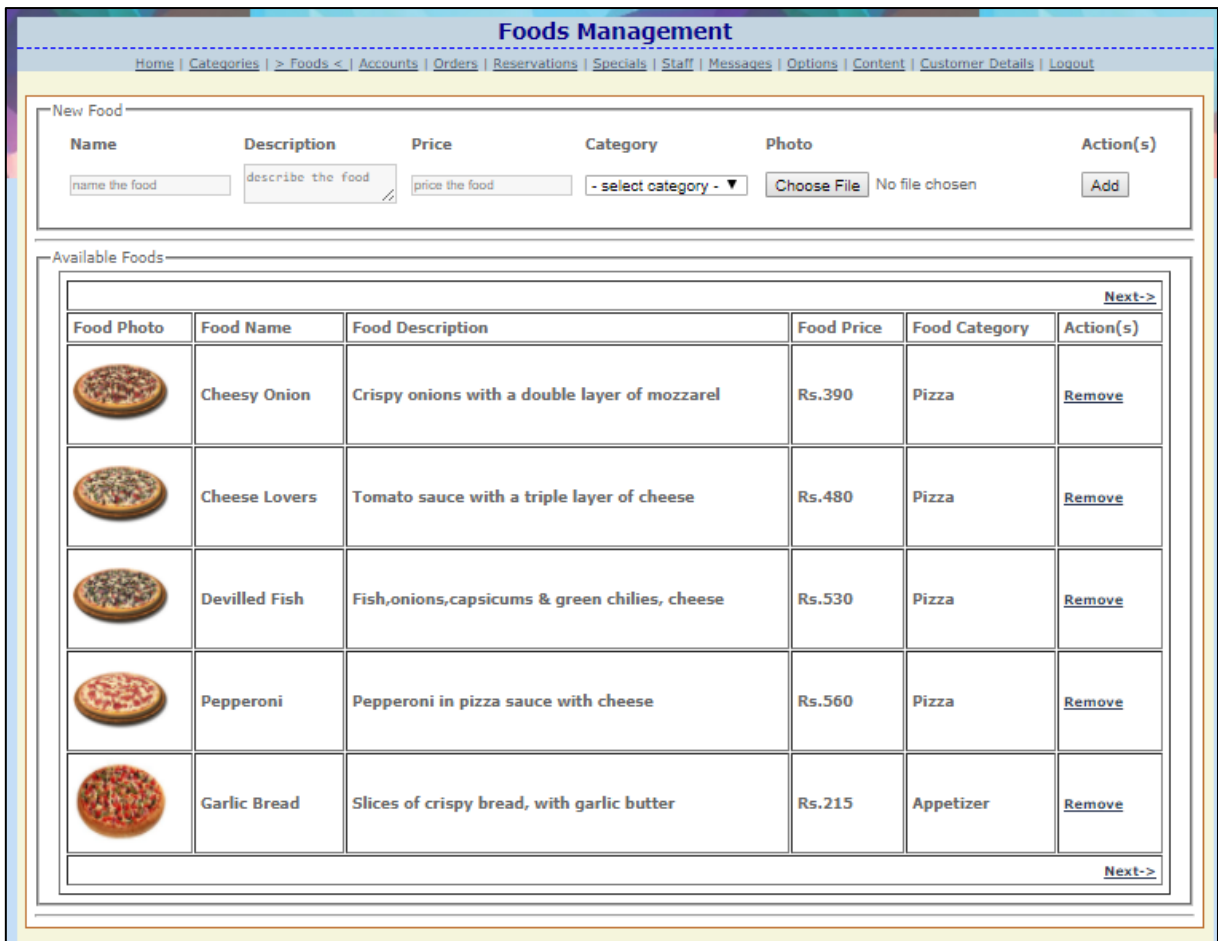


Figure E.4-add/remove food items

Figure E.5 shows the customers management interface of administrator login of the system.

The screenshot displays the 'Customers Management' interface. At the top, there is a navigation menu with links: Home | Categories | Foods | Accounts | Orders | Reservations | Specials | Staff | Messages | Options | Content | Customer Details | Logout. Below the menu, the main content area is titled 'Customers List'. It contains a table with the following data:

Customer ID	First Name	Last Name	Email	Action(s)
20	ppp	qqq	pqr@gmail.com	Remove
18	harshi	kumari	aaa@gmail.com	Remove
17	bbb	ccc	xyz@gmail.com	Remove
19	aaa	bbb	ddd@gmail.com	Remove
21	n	n	n@gmail.com	Remove
22	a	b	harshisubasinghe@gmail.co	Remove
23	a	b	harshisubasinghe@gmail.com	Remove

At the bottom of the page, there is a copyright notice: © 2017 PLATINUM PIER RESTAURANT. All Rights Reserved.

Figure E.5-veiw customer list

Appendix F – Program List

Below figure (figure F.1) shows the validation function of the admin login form.

```
//function to handle login-form validation
function loginValidate(loginForm){

    var validationVerified=true;
    var errorMessage="";

    if (loginForm.login.value=="")
    {
        errorMessage+="Username not filled!\n";
        validationVerified=false;
    }
    if(loginForm.password.value=="")
    {
        errorMessage+="Password not filled!\n";
        validationVerified=false;
    }
    if(!validationVerified)
    {
        alert(errorMessage);
    }
    return validationVerified;
}
```

Figure F.1-validation of admin login

Figure F.2 shows the program of login failed page. If the customer enters invalid user name or password, customer will direct to this page.

```
<div id="header">
    <div id="logo"> <a href="index.php" class="blockLink"></a></div>
    <div id="company_name"><?php echo $name ?></div>
</div>
<div id="center">
    <h1>Login Failed</h1>
    <div style="border:#bd6f2f solid 1px;padding:4px 6px 2px 6px">
<p>&nbsp;</p>
<div class="error">Login Failed!</div>
<p>Please check your email and password. <a href="login-register.php">Click Here</a> to try again.</p>
</div>
</div>
<div id="footer">
    <div class="bottom_menu"><a href="index.php">Home Page</a> | <a href="aboutus.php">About Us</a> |
    | <a href="admin/index.php" target="_blank">Administrator</a> |</div>

    <div class="bottom_addr">&copy; <?php echo date("Y") . " " . $name ?>. All Rights Reserved</div>
</div>
</div>
</body>
</html>
```

Figure F.2-login failed

Below figure (figure F.3) shows the validation function of the customer registration form.

```
//function to handle register-form validation
function registerValidate(registerForm){

var validationVerified=true;
var errorMessage="";

if (registerForm.fname.value=="")
{
errorMessage+="Firstname not filled!\n";
validationVerified=false;
}
if(registerForm.lname.value=="")
{
errorMessage+="Lastname not filled!\n";
validationVerified=false;
}
if (registerForm.login.value=="")
{
errorMessage+="Email not filled!\n";
validationVerified=false;
}
if(registerForm.password.value=="")
{
errorMessage+="Password not provided!\n";
validationVerified=false;
}
if(registerForm.cpassword.value=="")
{
errorMessage+="Confirm password not filled!\n";
validationVerified=false;
}
if(registerForm.cpassword.value!=registerForm.password.value)
{
errorMessage+="Password and Confirm Password do not match!\n";
validationVerified=false;
}
if (!isValidEmail(registerForm.login.value)) {
errorMessage+="Invalid email address provided!\n";
}
```

Figure F.3-register form validation

Figure F.4 shows the validation function of the table reservation.

```
var today = new Date();
var startTime = $('#sdate').val();
var inputDate = new Date(startTime);
var inputDate3 = inputDate.getTime();
var inputDate2 = today.getTime();
var inputDate1 = inputDate.getTime() - today.getTime();
var startime = $('#start').val();
var endTime = $('#end').val();

var startTimeArray = startTime.split(":");
var startInputHrs = parseInt(startTimeArray[0]);
var startInputMins = parseInt(startTimeArray[1]);

var endTimeArray = endTime.split(":");
var endInputHrs = parseInt(endTimeArray[0]);
var endInputMins = parseInt(endTimeArray[1]);

var startMin = startInputHrs * 60 + startInputMins;
var endMin = endInputHrs * 60 + endInputMins;

var result;

if (endInputHrs < startInputHrs) {
    alert("Invalid Time");
    return false;
}

if (endMin < startMin) {
    var minutesPerDay = 24 * 60;
    result = minutesPerDay - startMin; // Minutes till midnight
    result += endMin; // Minutes in the next day
} else {
    result = endMin - startMin;
}

if (inputDate.getTime() - today.getTime() <= -86400000) {
    alert("Invalid Date");
    return false;
}

var minutesElapsed = result % 60;
var hoursElapsed = (result - minutesElapsed) / 60;
if (hoursElapsed >= 2) {
    if (minutesElapsed > 0) {
        alert("maximum reservation time is 2 hours.");

        return false;
    }
    if (hoursElapsed > 2) {
        alert("maximum reservation time is 2 hours.");
        return false;
    }
} else {
    return true;
}
```

Figure F.4-validation of table reservation

Figure F.5 shows the program to send email for order confirmation once the customer confirm the order.

```
function Sendmail($order_id) {
    $mail_id = $_SESSION['SESS_EMAIL'];

    ini_set( 'display_errors', 1 );
    error_reporting( E_ALL );
    $from = 'platinumpier@gmail.com';
    $to = 'harshisubasinghe@gmail.com';
    $subject = 'Platinum Pier Restaurant oder Confirmation';
    $message = 'Dear customer,

Your order number is order no.' . $order_id.

'    Thank you for placing an order at Platinum Pier Restaurant. We are happy to inform that your
order has been received and will deliver your food with in one hour.

We love feedback, so if you have any suggestion for improvement or if there any question you would
like to address, feel free to contact us.

Thank you again for choosing Platinum Pier Restaurant.';

$headers = 'From:' . $from;
mail($mail_id,$subject,$message, $headers);
}
```

Figure F.5-order confirmation