

**DESIGN AND IMPLEMENTATION OF A HOTEL BOOKING
SYSTEM FOR WHITE CASTLE HOTEL**

BY

SANDALO WILBERFORCE

BIT/10123/81/DU

AND

TUMWEBAZE JAMES

BIT/10083/81/DU

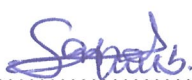
**A RESEARCH PROJECT SUBMITTED TO THE SCHOOL OF COMPUTER
STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR AWARD OF THE DEGREE OF BACHELOR OF
INFORMATION TECHNOLOGY OF KAMPALA
INTERNATIONAL UNIVERSITY**

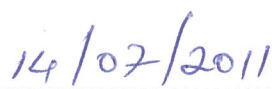
JULY, 2011

DECLARATION

We SANDALO WILBERFORCE and TUMWEBAZE JAMES hereby declare that this research project is our original work and its contents have not been published or submitted before for any academic award to any institution /university before.


SANDALO WILBERFORCE

Signature.....

Date.....

TUMWEBAZE JAMES

Signature.....

Date.....

APPROVAL

This is to certify that this project entitled *Design and Implementation of a Hotel Booking System* was conducted and written under my supervision.

ONKANGI C.

Signature.....

Date.....14/07/2011.....

DEDICATION

We dedicate this work to our parents ; kebikomi Joyce, Shyaka Isaac, Nantongo Jessica ,brothers; Rubyogo Wilson, Wangoli Oscar,Bwire Evans,Sandalo Jimmy,Muwanguzi Victor, and beloved sisters Nabwire Ritah,Nekesa Margaret,Nafula Christine,Anato for their moral support to see this success.

ACKNOWLEDGEMENT

Special acknowledgement to our beloved parents who have been so helpful to spiritually, moral upright and financial assistance.

We also do not forget to thank the staff of White Castle Hotel especially the manager for being open to us with good will assistance during the research survey.

My warm appreciation also goes to our supervisor Ms. Onkangi for having been so good towards my research.

The students and friends for their friendly help and assistance during the struggle.

Thank you so much.

LIST OF ACRONYMS

CRS	Central Reservation System
DBMS	Database Management System
MIS	Management Information system
RAID	Redundant Arrays of Independent Disks
TPS	Transaction Processing Systems
DFD	Data Flow Diagram
E-R	Entity Relation
GUI	Graphical User Interface

TABLE OF CONTENTS

DECLARATION.....	i
APPROVAL	ii
DEDICATION	iii
ACKNOWLEDGEMENT.....	iv
LIST OF ACRONYMS.....	v
LIST OF TABLES	xi
LIST OF FIGURES.....	xii
TABLE OF CONTENTS	vi
CHAPTER ONE.....	1
INTRODUCTION.....	1
1.0 Overview	1
1.1 Background of the Study.....	1
1.2 Statement of the Problem	2
1.3 Project Objectives.....	2
1.3.1 Main Objectives	2
1.4 Research Questions	2
1.5 Scope of the Study.....	2
1.6 Purpose of the Study.....	3
1.7 Significance of the Study	3
CHAPTER TWO.....	4
LITERATURE REVIEW	4
2.0 Introduction	4

2.1 Manual system.....	4
2.2 Database	5
2.3 Database Management System.....	5
2.4 Information System	7
2.4.1 Types of Information Systems	7
2.4.1.1 Transaction Processing Systems	7
2.4.1.2 Management Information Systems.....	8
2.4.1.3 Decision Support Systems.....	8
2.4.2 Advantages of an Information System	8
2.5 Booking System	9
2.6 Mobile Based System.....	9
2.6.1 Advantages of a mobile based system.....	9
2.7 Application of a Hotel Booking System.....	10
 CHAPTER THREE	 11
RESEARCH METHODOLOGY	11
3.0 Introduction	11
3.1 Research Techniques	11
3.2 Target Population	11
3.3 Area of Study.....	11
3.4 Sample and Sampling Technique	11
3.5 Data Collection Methods.....	12
3.5.1 Questionnaires	12
3.5.2 Interview.....	12
3.5.3 Document and Records Review	12

3.8 Data Analysis	13
CHAPTER FOUR	14
SYSTEM ANALYSIS AND DESIGN.....	14
4.0 Introduction.....	14
4.1 Data Model	14
4.2 Analysis of the Current System.....	14
4.3 Description of the Proposed System	15
4.4 Benefits of the Proposed System.....	15
4.5 Feasibility of Proposed System	16
4.5.1 Technical Feasibility	16
4.5.2 Operational Feasibility	16
Economical Feasibility	17
4.6 System Design.....	17
4.6.1 Conceptual Design	18
4.6.1.1 ER Diagram	18
4.6.1.2 Context Diagram	19
4.6.1.2 Data Flow Diagram	20
4.6.2 Physical Design	26
4.6.2.1 Tables	26
4.6.2.1.1 Accounts Table.....	26
4.6.2.1.2 Reservation Table.....	27
4.6.2.1.3 Users Table.....	28
4.6.2.1.4 Rooms Table.....	29
4.6.2.1.5 Room Type Table.....	30

4.6.2.1.6 Accounts Receivable Table	30
4.6.2.1.7 Business Info Table	31
4.6.2.1.8 Business Source Table.....	31
4.6.2.1.9 Charge Type Table	32
4.6.2.1.10 Customers Table.....	32
4.6.2.1.11 ID type Table.....	33
4.6.2.1.12 Rate per Period Table	33
4.6.2.1.13 User Permission Table.....	34
4.6.2.1.14 Room Status Table	34
4.6.2.1.15 Inventory Table	35
4.6.2.2 Forms	36
4.6.2.2.1 Reservation Form	36
4.6.2.2.2 Check In Form.....	37
4.6.2.2.3 Check Out Form	37
4.6.2.2.4 Amount Paid Form	38
4.7 Requirements Specification.....	38
4.7.1 System Requirements	38
4.7.1.1 Hardware Requirements	38
4.7.1.2 Software Requirements	39
4.8 System Testing	39
4.8.1 Software testing.....	39
4.9 System Implementation.....	40
4.9.1 Installation	40
4.9.2 System Conversion.....	40
4.9.3 User Training.....	41

4.9.4 Design Tools.....	41
CHAPTER FIVE.....	42
DISCUSSION, RECOMMENDATIONS AND CONCLUSION	42
5.0 Introduction	42
5.1 Discussion	42
5.2 Recommendations	42
5.3 Conclusion.....	42
5.4 Area of Future Research.....	43
5.5 Limitations of the Study	43
References	44
APPENDICE	45
Appendix 2: Questionnaire.....	93

LIST OF TABLES

Table 1: Economic feasibility of the new system.....	17
Table 2: Accounts Table.....	26
Table 3: Reservation Table.....	28
Table 4: Users Table.....	29
Table 5: Rooms Table.....	29
Table 6: Room Type table.....	30
Table 7: Accounts Receivable table.....	30
Table 8: Business Info table.....	31
Table 9: Business Source table.....	32
Table 10: Charge Type table.....	32
Table 11: Customers Table.....	33
Table 12: ID type Table.....	33
Table 13: Rate per Period Table.....	34
Table 14: User Permission Table.....	34
Table 15: Room Status Table.....	34
Table 16: Inventory Table.....	35

LIST OF FIGURES

Figure 1: ER diagram for the new system.....	18
Figure 2: Context Diagram for the new system.....	19
Figure 3: DFD of the check-in of the new customer.....	21
Figure 4: DFD of opening a new room.....	22
Figure 5: DFD for listing of the customers.....	23
Figure 6: DFD for checking -out customers.....	24
Figure 7: DFD showing bill of customers.....	25

CHAPTER ONE

INTRODUCTION

1.0 Overview

Today's hotel business management requires easy record system for the purposes of day to day operations, financial analysis and convenience to the customer.

Thousands of years have gone by in the universe and notably, each year has brought forth its own advancement with respect to data at hand, record handling methods and most importantly technology

This project focused on creating a computerized Hotel booking System where the customer simply calls in and the reservation arrangements are entered into an automated system. This was to relieve the hotel of the old file based system and spare the customer time because the customer doesn't have to be necessarily present to make reservations at the hotel.

1.1 Background of the Study

This research was carried out at the White Castle Hotel a local hotel operating in Kampala. White castle hotel offers hotel services to its customers. White castle hotel offers various accommodation packages to its customers.

White Castle Hotel is a mid-sized hotel that started in 2000. It offers accommodation to travelers in and outside Kampala and Uganda at large. Its customers have increased considerably since 2000 and consequently, it was difficult to efficiently track bookings and reliably meet the needs of the customers. The hotel sector in Uganda being a fast growing sector and very competitive in nature, to keep in business, White Castle Hotel has to meet the following conditions: be able to speedily serve customers, keep easily accessible information of bookings for management purposes and make customer bookings much easier and convenient.

White Castle Hotel decided to partially automate some of its departments and among them is the hotel booking department; therefore the researcher designed a computerized Hotel Booking System that makes booking easier.

1.2 Statement of the Problem

With the fast growing hotel business, attention of this leisure business is moving away from cutthroat competition to get customers but rather to better service provision to attract customers. As mobility in regard to mobile communication usage increases, there is need for improvement in the hotel business to catch up with this trend. This can be achieved by a realistic goal oriented mobile phone based hotel booking system.

1.3 Project Objectives

1.3.1 Main Objectives

The main objective of the study was to design a computerized booking system that manages hotel bookings and maintains records for management purposes.

1.3.2 Specific Objectives

1. To investigate the problems that arises from the current system.
2. To design a computerized booking system.
3. To test and implement the new computerized system.

1.4 Research Questions

1. What are the problems that arise from the current system?
2. What design can be used in the computerized booking system?
3. What strategies can be used in the new computerized system?

1.5 Scope of the Study

The Study was on designing a computerized Hotel Booking System for White Castle Hotel. The researcher took a period of four months.

1.6 Purpose of the Study

The purpose of the study was to design a computerized Hotel Booking System that enables the clients to make hotel reservations.

1.7 Significance of the Study

White Castle Hotel apparently uses a file based management system for managing bookings, so this project is intended to:

1. Keep track of the hotel bookings.
2. Store as much information as possible pertaining three major components, customer, account and branch
3. The system will also provide a reliable and easy reference to the history of the components.
4. Reduce unnecessary delays considering bookings can be only a call away.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

Literature review is an account of what has been published on a topic by credited scholars and researchers. The literature review will be collected from journals, books, internet magazines and articles.

2.1 Manual system

Manual reservation is a time consumable unlike when they have a computerized system. The disadvantages of manual hotel reservation system are; slow process of reservation, less data integrity, low in security and difficulty of searching and retrieving records.

Data duplication: the same data is repeated over and over since the workers find it hard to keep track of the documents, information and transactions

Lack of security: since data is stored in filing cabinets, it is freely available to anyone. If information falls into the wrong hands it can be used against, the hotel, customers, and can blackmail them.

Common errors: when entering data customers might have accidentally switched details and data since it is hand written.

Inconsistency of data: there will an unavailability of information for future use, since data might get misplaced during manual filing. So data will not be preserved properly for future use.

Repetition of work: if there are any changes to be made, the data will have to be entered again.

At times the worker would forget to make the changes or forget that they had already altered it and might redo it again, its again time consuming.

Too much paper work: since everything and every detail is written down manually in paper there will be too much paper work!

Space consuming: since the data and paper is stored in filing cabinets it consumes too much place, as the amount of work done on paper increases the filing cabinets too increases.

Slow retrieval of data: the information of customers and details are stored in different parts of the site and so takes a long time to retrieve the data. It takes a long time to find the information about

a relevant person. In case of a Delivery, the delivery will be held back. This results in a sharp drop in sales, unhappy customers and a bad impression on the hotel.

2.2 Database

A database is a collection of data or information organized for rapid search and retrieval, especially by a computer. Databases are structured to facilitate storage, retrieval, modification, and deletion of data in conjunction with various data-processing operations. A database consists of a file or set of files that can be broken down into records, each of which consists of one or more fields. Fields are the basic units of data storage. Users retrieve database information primarily through queries. Using keywords and sorting commands, users can rapidly search, rearrange, group, and select the field in many records to retrieve or create reports on particular aggregates of data according to the rules of the database management system (DBMS)

2.3 Database Management System

A collection of interrelated data together with a set of programs to access the data, also called database system, or simply database. The primary goal of such a system is to provide an environment that is both convenient and efficient to use in retrieving and storing information.

A database management system (DBMS) is designed to manage a large body of information. Data management involves both defining structures for storing information and providing relational, network, and hierarchical models. The latter two were widely used once, but are of declining importance. The relational model is very widely used. Databases based on the relational model are called relational databases.

The relational model uses a collection of tables (called relations) to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Each row of the table is called a tuple and each column represents the value of an attribute of the tuple.

The size of a database can vary widely, from a few megabytes for personal databases, to gigabytes (a gigabyte is 1000 megabytes) or even terabytes (a terabyte is 1000 gigabytes) for large corporate databases.

The information in a database is stored on a nonvolatile medium that can accommodate large amounts of data; the most commonly used such media are magnetic disks. Magnetic disks can store significantly larger amounts of data than main memory, at much lower costs per unit of data.

To improve reliability in mission-critical systems, disks can be organized into structures generically called redundant arrays of independent disks (RAID). In a RAID system, data are organized with some amount of redundancy (such as replication) across several disks. Even if one of the disks in the RAID system were to be damaged and lose data, the lost data can be reconstructed from the other disks in the RAID system.

Logically, data in a relational database are organized as a set of relations, each relation consisting of a set of records. This is the view given to database users. The underlying implementation on disk (hidden from the user) consists of a set of files. Each file consists of a set of fixed-size pieces of disk storage, called blocks. Records of a relation are stored within blocks. Each relation is associated with one or more files. Generally a file contains records from only one relation, but organizations where a file contains records from more than one relation are also used for performance reasons.

One way to retrieve a desired record in a relational database is to perform a scan on the corresponding relation; a scan fetches all the records from the relation, one at a time.

Accessing desired records from a large relation using a scan on the relation can be very expensive. Indices are data structures that permit more efficient access of records. An index is built on one or more attributes of a relation; such attributes constitute the search key. Given a value for each of the search-key attributes, the index structure can be used to retrieve records with the specified search-key values quickly. Indices may also support other operations, such as fetching all records whose search-key values fall in a specified range of values.

A database schema is specified by a set of definitions expressed by a data-definition language. The result of execution of data-definition language statements is a set of information stored in a special file called a data dictionary. The data dictionary contains metadata, that is, data about a database.

2.4 Information System

The organized collection, processing, transmission, and dissemination of information, in accordance with defined procedures, whether automated or manual. This includes the entire infrastructure, organization, and components that collect, process, store, transmit, display, disseminate, and act on information.

It's an integrated set of components for collecting, storing, processing, and communicating information. Business firms, other organizations, and individuals in contemporary society rely on information systems to manage their operations, compete in the marketplace, supply services, and augment personal lives. For instance, modern corporations rely on computerized information systems to process financial accounts and manage human resources; municipal governments rely on information systems to provide basic services to its citizens; and individuals use information systems to study, shop, bank, and invest.

2.4.1 Types of Information Systems

Information systems differ in their business needs. Three major information systems are:

2.4.1.1 Transaction Processing Systems

TPS processes business transaction of the organization. Transaction can be any activity of the organization. Transactions differ from organization to organization. For example, take a railway reservation system. Booking, canceling, etc are all transactions. Any query made to it is a transaction. However, there are some transactions, which are common to almost all organizations. Like employee new employee, maintaining their leave status, maintaining employees accounts, etc.

This provides high speed and accurate processing of record keeping of basic operational processes. These include calculation, storage and retrieval.

Transaction processing systems provide speed and accuracy, and can be programmed to follow routines functions of the organization.

2.4.1.2 Management Information Systems

These systems assist lower management in problem solving and making decisions. They use the results of transaction processing and some other information also. It is a set of information processing functions. It should handle queries as quickly as they arrive. An important element of MIS.

A database is a non-redundant collection of interrelated data items that can be processed through application programs and available to many users. And this will be the researcher's main focus.

2.4.1.3 Decision Support Systems

These systems assist higher management to make long term decisions. These type of systems handle unstructured or semi structured decisions. A decision is considered unstructured if there are no clear procedures for making the decision and if not all the factors to be considered in the decision can be readily identified in advance.

These are not of recurring nature. Some recur infrequently or occur only once. A decision support system must very flexible. The user should be able to produce customized reports by giving particular data and format specific to particular situations.

2.4.2 Advantages of an Information System

Better planning and control Management information systems have changed the dynamics of running businesses efficiently. Decentralization is one of the biggest advantages; it allows monitoring of operations at low levels and frees up resources for departmental managers to devote time to strategic activities. Coordination of specialized projects and activities is much better and decision makers in the organization are aware of issues and problems in all departments. Another advantage of MIS is that it minimizes information overload, which can be quite common with conventional businesses in the modern era.

Aid decision making MIS has to be designed and managed in such way that it aggregates information, monitors the company's activities and operations and enhances communication and collaboration among employees. This ensures better planning for all activities and better ways to measure performance, manage resources and facilitate compliance with industry and government regulations. Control helps in forecasting, preparing accurate budgets and providing the tools and vital information to employees, top management and business partners.

2.5 Booking System

Hotel reservations systems, commonly known as a central reservation system (CRS) are a computerized system that stores and distributes information of a hotel, resort, or other lodging facilities.

A Central Reservation System is a tool to reach the Global Distribution systems as well as Internet Distribution Systems from one single system, namely a central reservation system. A CRS is mainly an assistant for hoteliers to manage all of their online marketing and sales, where they can upload their rates & availabilities to be seen by all sales channels that are using a CRS. Sales Channels may include conventional travel agencies as well as online travel agencies. A hotelier using a central reservation system eases his/her tasks for online distribution, because a CRS does everything to distribute hotel information to the sales channels instead of the hotelier.

2.6 Mobile Based System

A mobile phone based hotel booking system is a system through which a customer can call in to the hotel and get the necessary arrangements of lodgings prepared to meet his /her needs without being present. Payments can always be effected later on.

2.6.1 Advantages of a mobile based system

The advantage is that usually transaction processing is really fast, it takes up usually a few seconds, however, if there a lot of files in queue, the time taken to process data might take a longtime. Another advantage is that it makes the process of booking fairer as files are processed in order in which they had been queued. (e.g. ticket booking)

2.7 Application of a Hotel Booking System

Creating a system to automate the services offered by White Castel Hotel, this would be more flexible than their earlier system. The features include: Storing, editing and reporting Staff Details, Storing, editing and reporting details of all the rooms available in the hotel and Managing details of customers' signing in & out of the hotel.

Input to the project; In Hotel Management System the following modules are to be entered. Input to staff details; Employee Name, Description of Employee's Post, Employee's Address, Employee's phone number, Joining Date Input for allotment of room to customers :Customer Name, Room No; Type-A/C or non-A/C, Customer's Balance Amount .Input to description of hotel rooms, Room Number, facility provided-A/C or non-A/C, Rent/day, Status of the room- Occupied / Unoccupied .Output of the project; Customized reports for the following: Details of various rooms and their status.

CHAPTER THREE

RESEARCH METHODOLOGY

3.0 Introduction

This chapter presents the methods and procedures used to collect and analyze data. It further looks at area of study, target population, sample size research design, sampling technique, data collection methods and data analysis.

3.1 Research Techniques

Qualitative means a non-numerical data collection or explanation based on the attributes of the graph or source of data. The researcher used quantitative research methods.

3.2 Target Population

The researcher collected the information from the manager, waitresses, customers, cashiers and supervisors. The questionnaires were distributed to the respondents and also interviewed during the process.

3.3 Area of Study

The research was carried out at the front desk booking office department of the White Castle Hotel. The research also covered all the necessary information about the people who call in booking and making reservation arrangements.

3.4 Sample and Sampling Technique

Sampling was part of statistical practice concerned with the selection of a subset of individuals from within a population to yield some knowledge about the whole population, especially for the purposes of making predictions based on statistical inference. There are various sampling techniques but the researcher used Cluster sampling because it was cheaper to 'cluster' the sample in some way e.g. by selecting respondents from certain areas only, or certain time-periods only.

This reduced travel and other administrative costs. It also meant that one does not need a sampling frame listing all elements in the target population. Instead, clusters were chosen from a cluster-level frame, with an element-level frame created only for the selected clusters. Cluster sampling generally increased the variability of sample estimates above that of simple random sampling, depending on how the clusters differ between themselves, as compared with the within-cluster variation.

3.5 Data Collection Methods

The researcher collected data from two sources; the primary and the secondary source. The primary source included questionnaire, interview and discussion. The secondary source was document and record review.

3.5.1 Questionnaires

The researcher distributed open questionnaires to managers, customers' cashiers and supervisors. The researcher used questionnaires because of the following reasons; gathered large amounts of information from large geographical divergent populations, respondents can remain anonymous and may be more willing to share a lot of information, there was less disruption at work and the respondents had ample time when responding to the questions.

3.5.2 Interview

The booking managers and customers were interviewed about the hotel's current booking system. The researcher used interviews because; it generally yields highest cooperation and lowest refusal rates, and there was higher response quality.

3.5.3 Document and Records Review

Records of the present booking system were gone through to analyze the records of previous bookings and reservations.

3.8 Data Analysis

The researcher analyzed data collected using SPSS program to verify the data into simple percentage which was presented in form of tables and charts. The researcher analyzed the data collected using questionnaires and interviews for accuracy and consistency in order to solve the problems of the existing system and meet the organizations needs.

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

4.0 Introduction

The chapter presents the requirements of the new system. The researcher evaluated the feasibility of the new system and looked at the real design and usage of the new system, the tools used to develop the system, processing and output in the design and implementation of a hotel booking system for White Castle Hotel.

4.1 Data Model

A data model is an integrated collection of concepts for describing and manipulating data, relationships between data and constraints on the data in an organization. In the design of this project, the relational data model was used. This model is based on the concept of mathematical notations. Data and relationships are represented as tables, each of which has a number of columns with unique names also known as attributes.

4.2 Analysis of the Current System

Currently, White Castle Hotel is using a manual file based booking system in information keeping and the other activities which is very hectic and not effective though but with soothing environment to spend quality time alone or with the loved ones as a desire of everyone. With White Castle hotel, you are sure of the right placement to enhance your relaxation to ward off fatigue from a long days stress.

The facilities include comfortable accommodation, bar and restaurants, banquet and meeting facilities and a 24 hour business centre. The spectacular White Castle Hotel is the perfect place to make a base for exploring Kampala city and its many attractions. The hotel is located in a convenient location in a scenic setting with striking views. All rooms and duplex suites are perfectly appointed and provided with digital satellite TV channels.

4.3 Description of the Proposed System

This system applies using mobile phone where users can book a hotel in the desired location. Even on the move, the customers can benefit from the system. Furthermore, the best price option means the cheapest offer can be determined with a single click.

All content on the mobile has been designed for use at a go. Once an appropriate list of hotels has been found, the room reservation can be completed conveniently using any mobile phone. Furthermore, mobile users can also use the portal to obtain information on hotel facilities and services, view photos. The simplified search function allows users to search for specific hotels or hotels in specific price categories. The service from the system will also be available as an iPhone application. This edition will include additional functions and have the special look & feel of these trendy mobile devices. Customers are Yahoo and eBay, media companies like MTN (Mobile Telephone Network), UTL (Uganda Telecom Limited), Orange as well as Warid.

The new system has a splash form after which the log in button with a User name, and password once an authorized user logs on, he/she is accesses the Main form. The main form has all the menu options that help the hotel to perform its daily transactions or functions through the reservation of the hotel.

4.4 Benefits of the Proposed System

Customers benefit in this respect from considerable cost and time savings. The system shows all available room rates for each hotel, thus always enabling customers to select the lowest or most appropriate rate. All bookings are forwarded simultaneously, securely and directly to the individual hotel.

Ensures increased functionality of the database. Every data will be entered in one database instead of there being a number of files, papers, books and registers scattered all over. This will create space that could be put to other use.

Help in providing data consistency thus improving accurate record keeping and data capture. Errors due to manual tracking will be reduced.

The system offers increased security to the company's system by use of user id's and passwords.

Only the relevant unit needs to enter information that is of importance to it. This reduces duplication of data. Information is easily shared therefore cutting on time wastage.

The development application and database ensures a unified system of record tracking and keeping.

4.5 Feasibility of Proposed System

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. “Feasibility Study” is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. .

4.5.1 Technical Feasibility

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. In technical feasibility the following issues were taken into consideration: whether the required technology was available or not and whether the required resources were available -manpower-programmers, testers & debuggers and software and hardware.

Once the technical feasibility was established, it was important to consider the monetary factors also. Since it could have happened that developing the online system could be technically possible but it could require huge investments and benefits could be less. For evaluating this, economic feasibility of the proposed system was carried out.

4.5.2 Operational Feasibility

Operational feasibility in Riaga’s, (2003), view is the determination of the new system will be able to perform the designated functions within the existing organizational environment with its current personnel and existing procedures. The system is put in use once it has been developed and implemented. In view of this project, operational feasibility is up to date since all the staff and top management wholly accept the new system to be developed. The original environment that encompasses of the personnel and staff members of White Castle Hotel who came up with the existing procedures played a big role in operational feasibility to help come up with the database system.

Economical Feasibility

Economic feasibility includes a broad range of concerns that includes cost-benefit analysis. In this project, the researcher weigh the cost and the benefits associated with the System and if it suits the basic purpose of the Hotel. The financial and the economic questions during the preliminary investigation are verified to estimate the following: the cost to conduct a full system investigation, the cost of hardware and software for the class of application being considered. This feasibility checks whether the system can be developed with the available funds. The Hotel Management System does not require enormous amount of money to be developed. This can be done economically if planned judiciously, so it is economically feasible. The cost of project depends upon the number of man hours required.

Item	Quantity	Cost	Total
Computer	1	700000	700000
Visual Basic 6.0	1	30000	30000
Ms Office 2003	1	30000	30000
Ms Server 2003	1	50000	50000
Transportation		100000	100000
Grand Total			910,000

Table 1: Economic feasibility of the new system

4.6 System Design

In designing the system, the researcher had to meet the business system requirements and constraints by translating them into a technical solution.

This involved the design of network architecture within departments, database to store information, and user interfaces in order to meet the user's requirements. The researcher considered the conceptual design and physical design

4.6.1 Conceptual Design

This is concerned with the conversion of logical records structures of a data model supported by a database management system, identifying entities and their matching attributes and the relationship types determining the attributes domain.

4.6.1.1 ER Diagram

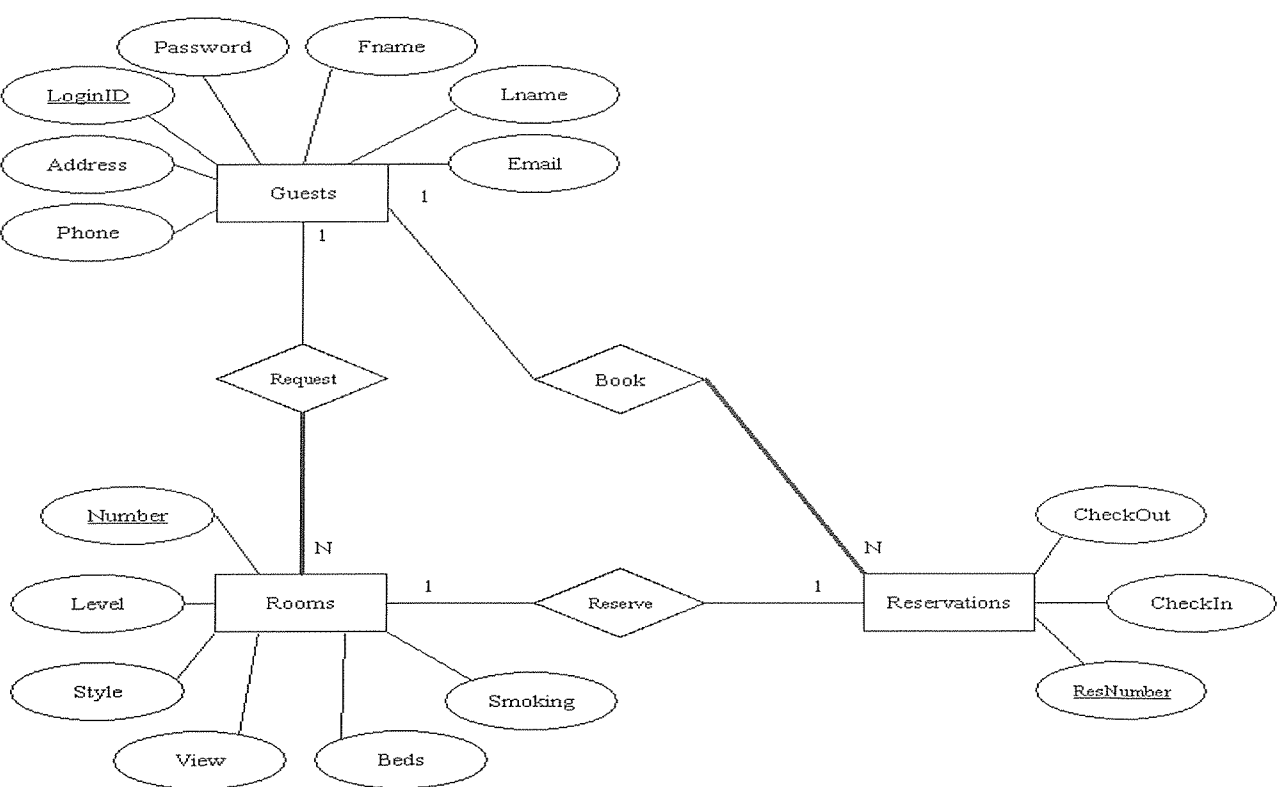


Figure 1: ER diagram for the new system

4.6.1.2 Context Diagram

CONTEXT LEVEL DFD
HOTEL MANAGEMENT SYSTEM

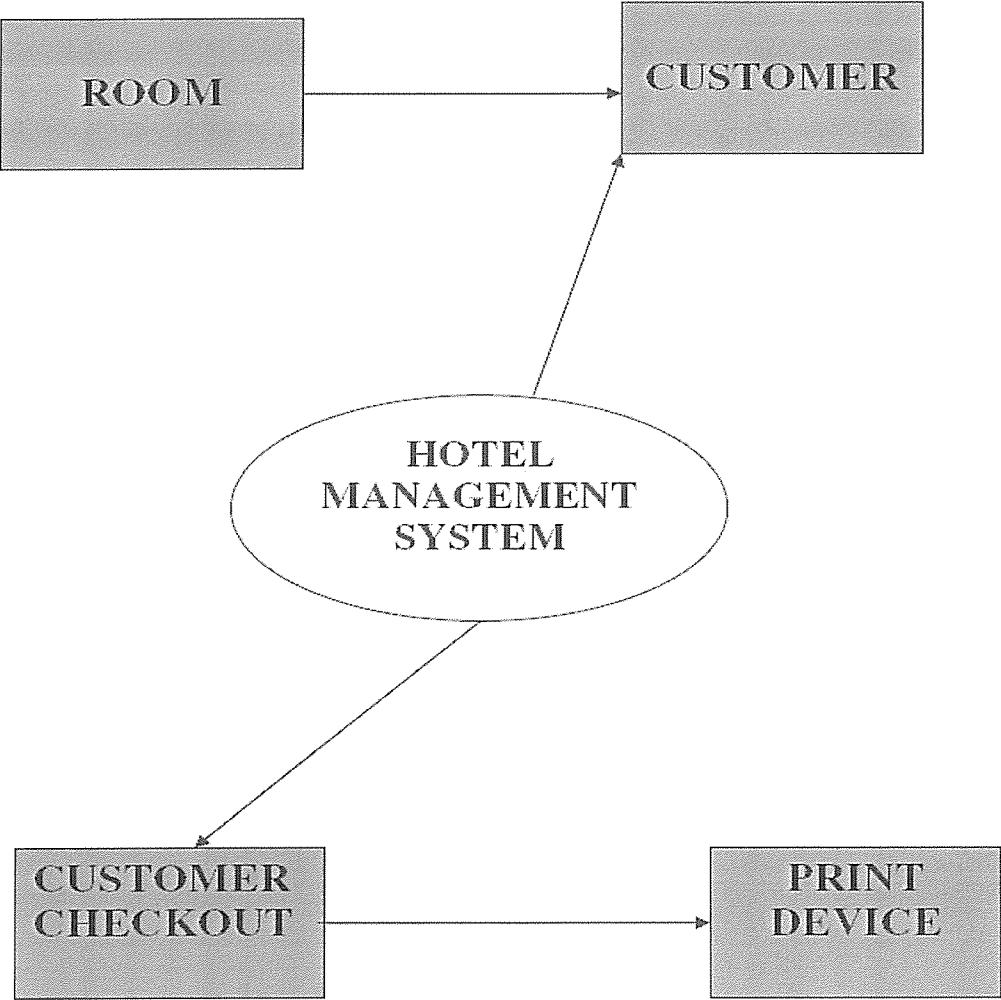


Figure 2: Context Diagram for the new system

4.6.1.2 Data Flow Diagram

A DFD is a pictorial representation of the path which data takes from its initial interaction with the existing system until it completes any interaction. The diagram will describe the logical data flows dealing the movements of any physical items. The DFD also gives the insight into the data that is used in the system i.e., who actually uses it is temporarily stored. A DFD does not show a sequence of steps. A DFD only shows what the different sequence of steps. A DFD only shows what the different process in a system is and what data flows between them. The following are some DFD symbols used in the project

DATA FLOW DIAGRAM

CHECK-IN OF A NEW CUSTOMER

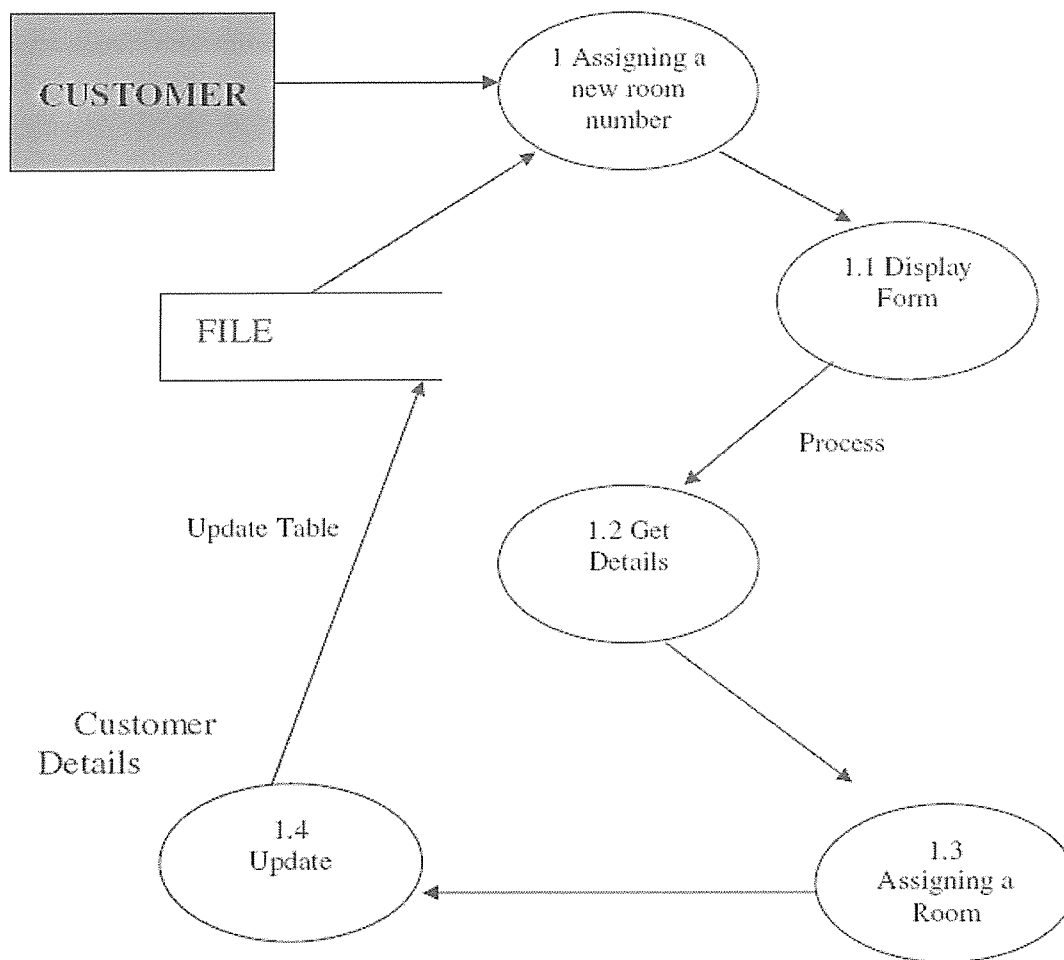


Figure 3: DFD of the check-in of the new customer

DATA FLOW DIAGRAM OPENING A NEW ROOM

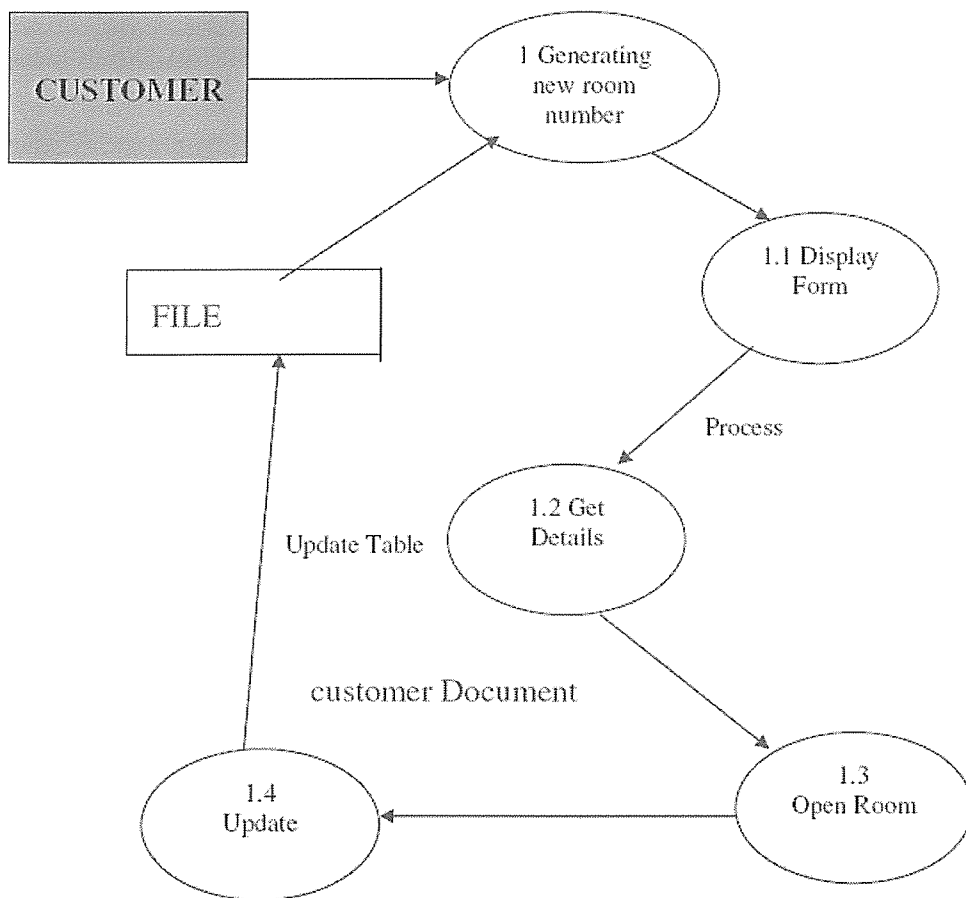


Figure 4: DFD of opening a new room

DATA FLOW DIAGRAM LISTING OF CUSTOMERS

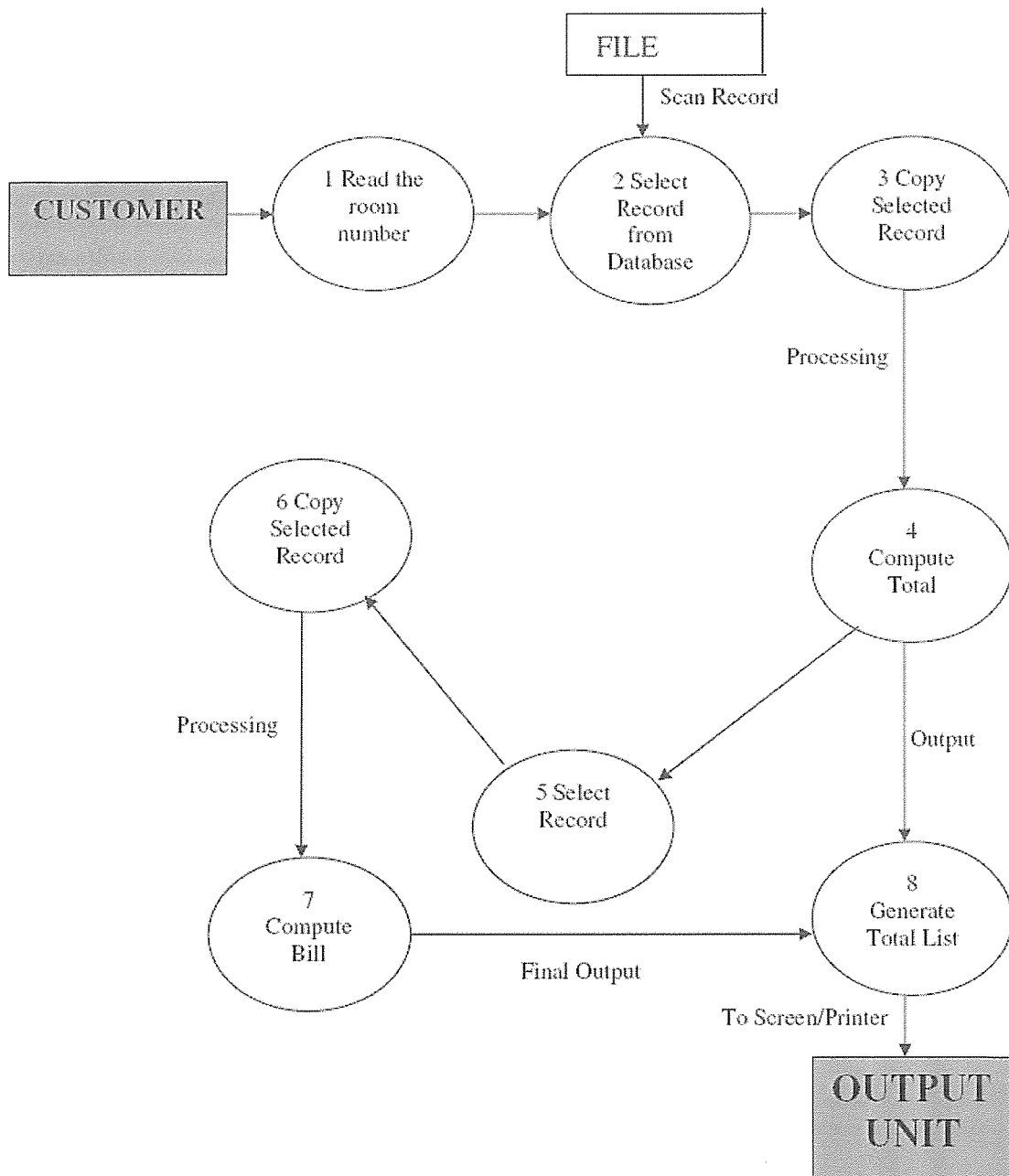


Figure 5: DFD for listing of the customers

DATA FLOW DIAGRAM CHECK-OUT OF CUSTOMER

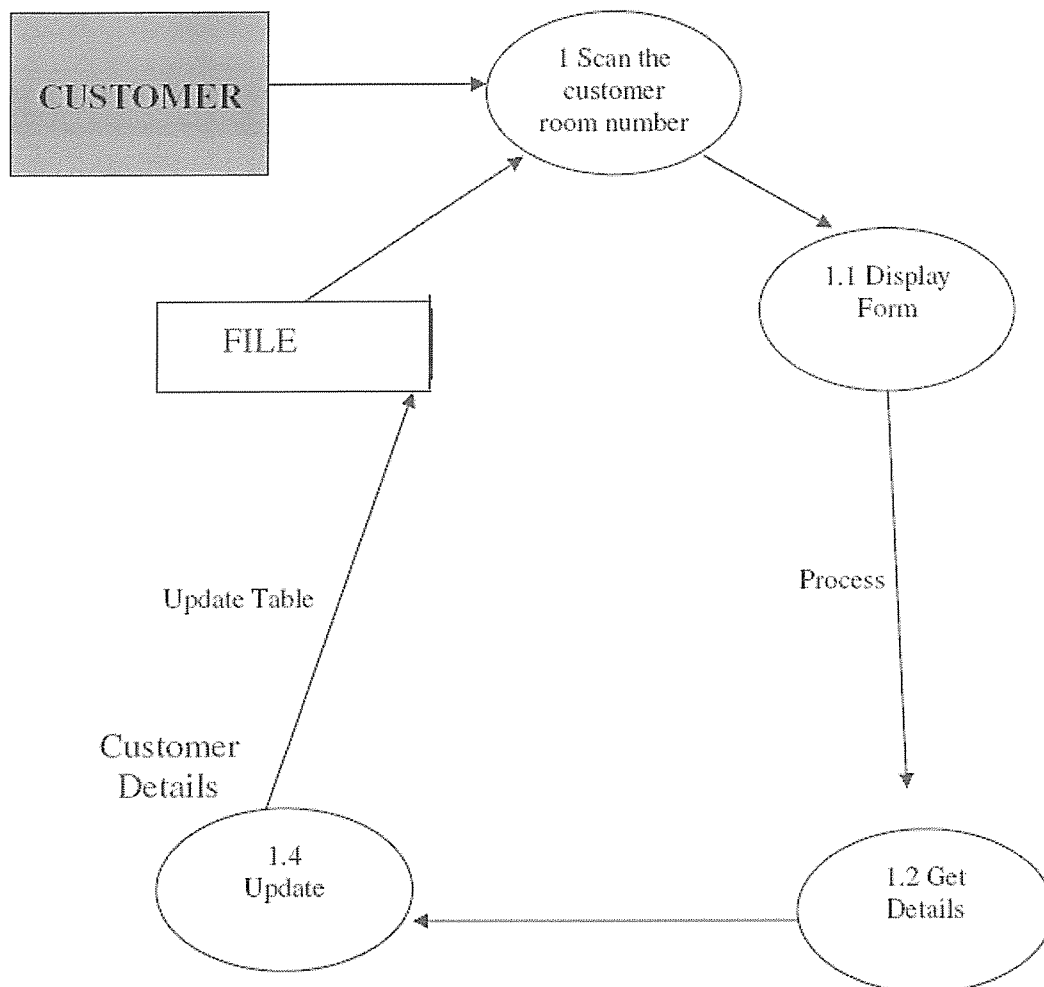


Figure 6: DFD for checking -out customers

DATA FLOW DIAGRAM

GENERATING BILL OF CUSTOMER

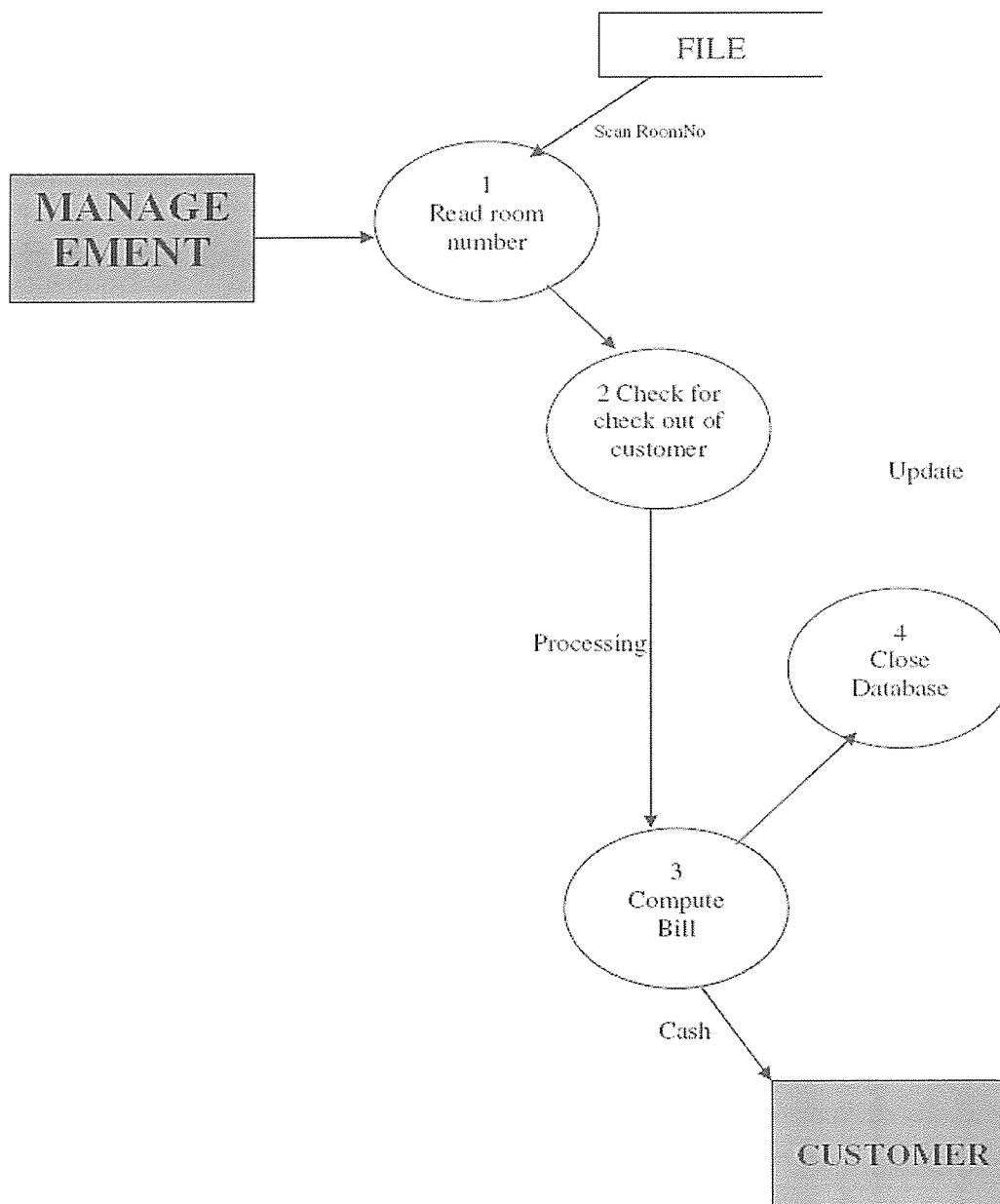


Figure 7: DFD showing bill of customers

4.6.2 Physical Design

4.6.2.1 Tables

4.6.2.1.1 Accounts Table

Fields	Type	Description
Account code	Text	A unique identifier for the employee
Fullnames	Text	full name of the employee
Position	Text	position of the employee at the Hotel
Security level	Number	denotes the authority to access corporate files
Login name	Text	login name of employee for the system
Password	Text	password for system
Date added	Date/Time	
Added By Fk	Number	
Date modified	Date/Time	
Last user Fk	Number	

Table 2: Accounts Table

4.6.2.1.2 Reservation Table

Fields	Type	Description
Reservation No	Text	
CustomerID	Number	
Address	Text	
CountryID	Number	
CompanyID	Number	
IDTypeID	Number	
IDNumber	Text	
RoomNumber	Number	
DateIn	Date/Time	
Dateout	Date/Time	
RateType	Number	
Rate	Currency	
OtherCharges	Currency	
Discount	Currency	
AmountPaid	Currency	
Days	Number	
Adults	Text	
Childrens	Text	

Total	Currency	
BusinessSourceID	Number	
VehicleID	Number	
VehicleModel	Text	
PlateNo	Text	
Status	Text	
Notes	Text	
ReservedBy	Text	
CancelledBy	Text	
DateAdded	Date/Time	
AddedByFK	Number	
Date Modified	Date/Time	
Last UserFK	Number	
Date Added	Date/Time	
Added ByFK	Number	

Table 3: Reservation Table

4.6.2.1.3 Users Table

Fields	Type	Description
PK	Number	
User ID	Text	

Password	Text	
Complete Name	Text	
Admin	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 4: Users Table

4.6.2.1.4 Rooms Table

Fields	Type	Description
Room Number	Number	Room Number
Room Type ID	Number	1=single with dstv etc.
Room Status ID	Number	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 5: Rooms Table

4.6.2.1.5 Room Type Table

Fields	Type	Description
Room Type ID	AutoNumber	
Room Type	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 6: Room Type table

4.6.2.1.6 Accounts Receivable Table

Fields	Type	Description
Acc Rec ID	AutoNumber	Account Receivable ID
Company ID	Number	
Folio Number	Text	
Debit	Currency	
Credit	Currency	

Table 7: Accounts Receivable table

4.6.2.1.7 Business Info Table

Fields	Type	Description
Name	Text	
Address	Text	
Contact Info	Text	

Table 8: Business Info table

4.6.2.1.8 Business Source Table

Fields	Type	Description
Business Source ID	AutoNumber	
Business Source	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 9: Business Source table

4.6.2.1.9 Charge Type Table

Fields	Type	Description
Charge Type ID	AutoNumber	
Charge Type	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 10: Charge Type table

4.6.2.1.10 Customers Table

Fields	Type	Description
Customer ID	Number	
Last Name	Text	
First Name	Text	
Address	Text	
Email Address	Text	
Tel No	Text	
Mobile No	Text	
Fax No	Text	
Date Added	Date/Time	

Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 11: Customers Table

4.6.2.1.11 ID type Table

Fields	Type	Description
ID Type ID	AutoNumber	
ID Type	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 12: ID type Table

4.6.2.1.12 Rate per Period Table

Fields	Type	Description
Folio Number	Text	
Date	Date/Time	
Room Number	Number	
Rate Type ID	Number	

Rate	Currency	
Adults	Currency	
Childrens	Currency	

Table 13: Rate per Period Table

4.6.2.1.13 User Permission Table

Fields	Type	Description
User Permission ID	AutoNumber	
User ID	Text	
Form ID	Number	
Allow Open	Yes/No	

Table 14: User Permission Table

4.6.2.1.14 Room Status Table

Fields	Type	Description
Status ID	AutoNumber	
Status	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 15: Room Status Table

4.6.2.1.15 Inventory Table

Fields	Type	Description
Status ID	AutoNumber	
Status	Text	
Date Added	Date/Time	
Added By FK	Number	
Date Modified	Date/Time	
Last User FK	Number	

Table 16: Inventory Table

4.6.2.2 Forms

4.6.2.2.1 Reservation Form

Reservation

Reservation

Fill all fields or fields with "" then click 'Save' button to update.

Guest Information

Reservation No.08-00055

Last nameYbanez

First nameVicente

Address

CountryPhilippines

Company

Identification Information

ID TypeDriver License

ID Number

Vehicle Information

Vehicle Make

Model

Plate No.

Modification History

Rate Information

Room Number204 - Standard Room

Date In6 / 4 /2008

Rate TypeDouble

No. of Days1

No. of Adults2

No. of Childrens0

Business SourceReservation

Notes

Date Out6 / 5 /2008

Rate/Period998.00

Total Charges998.00

Other Charges0.00

Sub Total998.00

Discount0.00 %

Total998.00

Amount Paid0.00

Balance998.00

Save

Close

Reserved By:

Cancelled By:

Business Source:

Figure 8: Reservation Form

36

4.6.2.2.2 Check In Form

Check In
Fill all fields or fields with "*" then click 'Save' button to update.

Guest Information

Folio Number: 08-00204
 R Card No.:
 Last Name: Pabalinas
 First name: Jein
 Address:
 Country: Philippines
 Company:

Rate Information

Room Number: 209
 Date In: 6 / 4 / 2008
 Rate Type: Double
 No. of Days: 1
 No. of Adults: 2
 No. of Childrens: 0
 Business Source: Walk-In

Identification Information

ID Type: Driver License
 ID Number:

Vehicle Information

Vehicle Make:
 Model:
 Plate No.:

Summary Table:

Date Out	6 / 5 / 2008
Rate/Period	1,458.00
Total Charges	1,458.00
Other Charges	0.00
Sub Total	1,458.00
Discount	0.00 %
Total	1,458.00
Amount Paid	0.00
Balance	1,458.00

Buttons: Charge Room, Check In, Update, Cancel

Footer: Reserved By, Check In By, Check Out By, Business Source:

Figure 9: Check In form

4.6.2.2.3 Check Out Form

Check Out
Fill all fields or fields with "*" then click 'Save' button to update.

Room Number: 203
 Guest Name: Alfie Fajura

Date In: 5/29/2008
 Rate Type: Double
 No. of Days: 7
 No. of Adults: 2
 No. of Childrens: 0

Date Out: 6 / 5 / 2008

Summary Table:

Rate/Period	1,458.00
Total Charges	10,206.00
Other Charges	0.00
Sub Total	10,206.00
Discount	0.00 %
Total	10,206.00
Amount Paid	0.00
Balance	10,206.00

Buttons: Print, Check Out, Cancel

Figure 10: Check Out Form

4.6.2.2.4 Amount Paid Form

Amount Paid Details

Payments

Fill all fields or fields with "*" then click 'Save' button to update.

Guest Name

Alfie Fajura

Date

4 / 1 / 2008

Balance

0.00

Amount Paid

Payment Method

Payment Type

Cash

Card Number

Auth Code

Valid Till

Add

Update

	Date	Payment Type	Card Number	Auth Code	Valid Till	Amount Paid
1	4 / 1 / 2008	Cash				10,205.00

Close

Figure 11: Amount Paid Form

4.7 Requirements Specification

This was intended in identifying and defining the requirements of the new system based on the findings, during the analysis activities on the existing system. User requirements study was conducted through establishing the data flows of the existing system operations, interviewing the different users provided a general overview of what and how existing system does.

4.7.1 System Requirements

These requirements specify the minimum hardware and software specifications necessary for the system to run efficiently on any given computer platform / system.

4.7.1.1 Hardware Requirements

The hardware requirements for the system to run smoothly are:-

A personal computer which hosts the database must be available. The minimum specifications of the computer should be:

1. Central Processing Units (CPU) with at least a Pentium IV processor, hard disk size of at least 40GB and RAM size of at least 512MB
2. A graphics card and monitor capable of supporting a resolution of at least 800X600 pixels
3. A CD-ROM drive for software installations.
4. Printers to print out the reports.

4.7.1.2 Software Requirements

The system will be developed using the following.

1. Microsoft Access Database(Back-end)
2. Microsoft Visual basic6.0(Front-end)
3. Microsoft Windows server(Operating system)
4. Microsoft office2000 and above
5. Antivirus software to detect and disinfect viruses that may affect stored records.

4.8 System Testing

4.8.1 Software testing

This is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing was used to provide an objective, independent view of the software to allow the hotel to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding errors or other defects. System testing was also done to make sure the system performs as the requirements and specifications it should perform. It was to cover the overall requirements and specifications, and, all combined parts of the system.

4.9 System Implementation

The researcher's decision to implement the Hotel Management system using standard PC technology and a commercial database product has narrowed our choices of implementation algorithms and data structures. We have chosen to implement this system with a PC database program (Access) that uses a relational data model. Basically what this means is that related data is grouped together in what can be logically viewed as a columnar table of records. Each record within a particular table is composed of a number of fields that hold the relevant information. Each record can be uniquely described by one or more fields in the record. These fields are referred to as the records primary key. Information is retrieved from the table by specifying the records primary key. The entire system is made up on a number of tables, each of which can be thought of as describing one logical entity. For example a customer is represented by a table that has field entries corresponding to the customer's name, address and many more. Information about each customer is retrieved by specifying the customer's name which is the primary key for the customer table.

With this basic information in mind we have specified the required tables, their primary keys and related field information below. Each table is laid out as a heading, followed by a list of the fields that make up the table.

4.9.1 Installation

The front desk and the management department's computers had Windows XP installed on them. Then Microsoft office was also installed on the machines along with Visual Basic 6.0. The reservation system software was also installed on these machines and a peer-peer network was put in place to make sure that these two departments share this resource concerning reservations and management of the hotel. Double click the setup, the auto run will initialize the system and will start installation in which different and several options are prompted where the one installing will be prompted to "next, continue or I accept" until to completion (Finish). The system was successfully installed at the Hotel.

4.9.2 System Conversion

The old file based system had to be checked to see if the new system customarily covers all the needs of the booking department and also to fine tune it to the best of White castles needs.

Conversion into new system from old system was a significant milestone. This included a strategy for converting from the old system to the new system. It is highly recommended that the pre-current system keeps running along with this one parallel to comprehend effective acquisition of the purpose and objective for which it was developed. The parallel conversion will minimize the likely errors when using it and thus allows room for debugging and fixing those sub sequential errors.

4.9.3 User Training

After the system was converted to the server of White Castle Hotel, the system designer had to train the system users on how to operate the system. First advised the users not to share their passwords. The designer also trained the manager because was the one who was going to act as the system administrator and later train the other departments.

4.9.4 Design Tools

The need of hotel software development is competent in a GUI based front-end tool, which can connect to Relational Database engines. This gave the researcher the opportunity to develop client server based commercial applications. The researcher used Microsoft Access Database (Back-end) and Microsoft Visual Basic 6.0(Front-end).

CHAPTER FIVE

DISCUSSION, RECOMMENDATIONS AND CONCLUSION

5.0 Introduction

This chapter presents the conclusion and recommendations that the researcher made to White Castle Hotel about the design and implementation of a Hotel Booking System. It further presents, the area of future research, and the limitations of the study

5.1 Discussion

The main objectives of the research were achieved through the reservation system for White Castle Hotel limited. However the system needed to be extended to cover all the departments of White Castle Hotel and its various branches country wide. White Castel Hotel put in place a local area network to cover all the departments of the hotel and a wide area network to cover all the various branches of the hotel.

5.2 Recommendations

The researcher recommends the following in order to improve on the existing system:

Employees (those working in the records and some staff authority to these details about students) should be taught how to use the system.

Passwords should be made private to those are supposed to use them, for the purpose of security.

The hotel should also ensure a backup is done periodically.

5.3 Conclusion

The main objective of the researcher was to solve the problem of the existing system by developing a new and reliable system. This was achieved by the researcher giving clear and precise information about the system in the conceptual model and later by the research activities that aided the full research work.

5.4 Area of Future Research

Since the system is under operation, the researcher will be involved in system support which includes the following activities:

- i. Program maintenance
- ii. Technical support
- iii. System recovery and backup

5.5 Limitations of the Study

The researcher found the following limitations;

The system could not be placed on all of White Castle Hotel branches because they were far apart or spread over the country far apart.

There was no network in place to cover all of White Castle Hotels different departments to enable these departments share this resource.

Some information concerning the hotels day to day operations was denied to the researcher because management reasoned that it was sensitive.

References

Codd, E.F. (1970), "*A relational Model of Data for Large Shared Data Banks*", In: Communications of the ACM 13 (6): 377–387.

Development of an object-oriented DBMS; Portland, Oregon, United States; Pages: 472 - 482; 1986; *Performance enhancement through replication in an object-oriented DBMS*; Pages 325-336;

Seltzer, M. (2008, July). *Beyond Relational Databases*. *Communications of the ACM*, 51(7), 52-58. Retrieved July 6, 2009, from Business Source Complete database.

Integration Definition for Information Modeling (IDEFIX) 21 December 1993.

Stephen Chu, in Conrick, M. (2006), *Introducing databases Health informatics: transforming healthcare with technology*, Thomson, p. 69.

"*Routine Database Maintenance*". PostgreSQL.org.
<http://www.postgresql.org/docs/8.3/static/routine-vacuuming.html>. Retrieved 2010-08-29.

"*Design of Main Memory Database System/Overview of DBMS*". En.wikibooks.org.
http://en.wikibooks.org/wiki/Design_of_Main_Memory_Database_System/Overview_of_DBMS. Retrieved 2010-08-29.

Ken Black (2004). *Business Statistics for Contemporary Decision Making* (Fourth (Wiley Student Edition for India) ed.). Wiley-India.

Scott, A.J., and Wild, C.J. (1986). *Fitting logistic models under case-control or choice-based sampling*. J. Roy. Statist. Soc. B, 48, 170–182.

APPENDICE

Appendix 1: Program Code

FrmAccountReceivable(code)

Public PK As Long

Public Company As String

Public Balance As Currency

Public AmountPaid As Currency

Dim clRowCount As Integer

Dim Amount As Currency

Private Sub btnRemove_Click()

 'Remove selected load product

 With Grid

 'Update the record count

 clRowCount = clRowCount - 1

 If .Rows = 2 Then Grid.Rows = Grid.Rows + 1

 .RemoveItem (.RowSel)

 End With

```
btnRemove.Visible = False
```

```
txtBalance.Text = toMoney(toNumber(txtBalance.Text) + toNumber(Grid.TextMatrix(Grid.RowSel,  
5)))
```

```
txtAmountPaid.Text = toMoney(txtAmountPaid.Text) - toNumber(Grid.TextMatrix(Grid.RowSel, 5))
```

```
Grid_Click
```

```
End Sub
```

```
Private Sub cmdAdd_Click()
```

```
    If Trim(txtAmount.Text) = "0.00" Then Exit Sub
```

```
    If toNumber(txtAmount) > Balance Then
```

```
        MsgBox "Amount paid exceed balance. Please enter correct amount.", vbInformation
```

```
        txtAmount.SetFocus
```

```
    Exit Sub
```

```
End If
```

```
Dim CurrRow As Integer
```

```
'Add to grid
```

```
With Grid
```

'Perform if the record is not exist

If .Rows = 2 And .TextMatrix(1, 2) = "" Then

.TextMatrix(1, 2) = dtpDate.Value

.TextMatrix(1, 3) = dcPaymentType.BoundText

.TextMatrix(1, 4) = dcPaymentType.Text

.TextMatrix(1, 5) = toMoney(txtAmount.Text)

Else

.Rows = .Rows + 1

.TextMatrix(.Rows - 1, 2) = dtpDate.Value

.TextMatrix(.Rows - 1, 3) = dcPaymentType.BoundText

.TextMatrix(.Rows - 1, 4) = dcPaymentType.Text

.TextMatrix(.Rows - 1, 5) = toMoney(txtAmount.Text)

.Row = .Rows - 1

End If

'Increase the record count

cIRowCount = cIRowCount + 1

txtBalance.Text = toMoney(Balance - toNumber(txtAmount.Text))

' Balance = Balance - toNumber(txtAmountPaid.Text)

```
txtAmountPaid.Text = toMoney(toNumber(txtAmountPaid.Text) + toNumber(txtAmount.Text))
```

```
'Highlight the current row's column
```

```
.ColSel = 5
```

```
'Display a remove button
```

```
Call Grid_Click
```

```
Call ResetFields
```

```
End With
```

```
End Sub
```

```
Private Sub cmdCancel_Click()
```

```
Unload Me
```

```
End Sub
```

```
Public Sub cmdSave_Click()
```

```
Dim rsPayments As New Recordset
```

```
rsPayments.CursorLocation = adUseClient
```

```
rsPayments.Open "SELECT * FROM [Accounts Receivable] WHERE AccRecID=" & PK, CN,  
adOpenStatic, adLockOptimistic
```

```
With rsPayments
```

```
.Fields("Debit") = txtAmountPaid.Text
```

```
.Update
```

```
End With
```

```
DeleteItems
```

```
Dim rsPaymentRec As New Recordset
```

```
rsPaymentRec.CursorLocation = adUseClient
```

```
rsPaymentRec.Open "SELECT * FROM [Payments Received] WHERE AccRecID=" & PK, CN,  
adOpenStatic, adLockOptimistic
```

```
Dim c As Integer
```

```
With Grid
```

```
'Save the details of the records
```

```
For c = 1 To cIRowCount
```

```
.Row = c
```

```
If .TextMatrix(c, 1) = "" Then
```

```
rsPaymentRec.AddNew
```

```
rsPaymentRec![AccRecID] = PK
```

Else

rsPaymentRec.Filter = "RecievedPaymentID = " & toNumber(.TextMatrix(c, 1))

' If rsPaymentRec.RecordCount = 0 Then

' rsPaymentRec.AddNew

'

' rsPaymentRec![AccRecID] = PK

' End If

End If

rsPaymentRec![Date] = .TextMatrix(c, 2)

rsPaymentRec![PaymentTypeID] = .TextMatrix(c, 3)

rsPaymentRec![Amount] = .TextMatrix(c, 5)

rsPaymentRec.Update

Next c

End With

'Clear variables

c = 0

Set rsPayments = Nothing

Set rsPaymentRec = Nothing

Unload frmAccountReceivable

End Sub

Private Sub cmdUpdate_Click()

If Trim(txtAmount.Text) = "0.00" Then Exit Sub

txtBalance.Text = toMoney(txtBalance.Text) + Amount

txtAmountPaid.Text = toMoney(txtAmountPaid.Text) - Amount

txtBalance.Text = toMoney(toNumber(txtBalance.Text) - toNumber(txtAmount.Text))

txtAmountPaid.Text = toMoney(toNumber(txtAmountPaid.Text) + toNumber(txtAmount.Text))

With Grid

.TextMatrix(.RowSel, 2) = dtpDate.Value

.TextMatrix(.RowSel, 3) = dcPaymentType.BoundText

.TextMatrix(.RowSel, 4) = dcPaymentType.Text

.TextMatrix(.RowSel, 5) = toMoney(txtAmount.Text)

End With

Call Grid_Click

Call ResetFields

End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then

SendKeys "{TAB}"

End If

End Sub

Private Sub Form_Load()

Call InitGrid

bind_dc "SELECT * FROM [Payment Type]", "PaymentType", dcPaymentType, "paymentTypeID",
True

txtCompany.Text = Company

txtBalance.Text = toMoney(Balance)

txtAmountPaid.Text = toMoney(AmountPaid)

dtpDate.Value = Date

DisplayForEditing

If Balance = "0.00" Then

Grid.Height = 3000

Grid.Top = 2160

dtpDate.Visible = False

dcPaymentType.Visible = False

txtAmount.Visible = False

cmdAdd.Visible = False

cmdUpdate.Visible = False

lblPaid.Visible = True

End If

End Sub

Procedure used to initialize the grid

Private Sub InitGrid()

clRowCount = 0

With Grid

.Clear

.ClearStructure

.Rows = 2

.FixedRows = 1

.FixedCols = 1

```
.Cols = 6
```

```
.ColSel = 5
```

```
'Initialize the column size
```

```
.ColWidth(0) = 315
```

```
.ColWidth(1) = 0
```

```
.ColWidth(2) = 1200
```

```
.ColWidth(3) = 0
```

```
.ColWidth(4) = 1200
```

```
.ColWidth(5) = 1200
```

```
'Initialize the column name
```

```
.TextMatrix(0, 0) = ""
```

```
.TextMatrix(0, 1) = "RecievedPaymentID"
```

```
.TextMatrix(0, 2) = "Date"
```

```
.TextMatrix(0, 3) = "Payment Type ID"
```

```
.TextMatrix(0, 4) = "Payment Type"
```

```
.TextMatrix(0, 5) = "Amount"
```

```
End With
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    frmAccountReceivableList.RefreshRecords
```

```

Set frmAccountReceivable = Nothing

End Sub

Private Sub Grid_Click()

    With Grid

        dtpDate.Value = .TextMatrix(.RowSel, 2)

        dcPaymentType.BoundText = .TextMatrix(.RowSel, 3)

        txtAmount.Text = toMoney(.TextMatrix(.RowSel, 5))

        Amount = .TextMatrix(.RowSel, 5)

        If Grid.Rows = 2 And Grid.TextMatrix(1, 2) = "" Then

            btnRemove.Visible = False

        Else

            btnRemove.Visible = True

            btnRemove.Top = (Grid.CellTop + Grid.Top) - 20

            btnRemove.Left = Grid.Left + 50

        End If

    End With

End Sub

```

```
Private Sub ResetFields()
```

```
    txtAmount.Text = ""
```

```
    txtAmount.SetFocus
```

```
End Sub
```

```
Private Sub DisplayForEditing()
```

```
    On Error GoTo err
```

```
    'Display the details
```

```
    Dim rsPayments As New Recordset
```

```
    clRowCount = 0
```

```
    rsPayments.CursorLocation = adUseClient
```

```
    rsPayments.Open "SELECT * FROM [qry_Payment_Received] WHERE AccRecID=" & PK, CN,  
adOpenStatic, adLockOptimistic
```

```
    If rsPayments.RecordCount > 0 Then
```

```
        rsPayments.MoveFirst
```

```
        While Not rsPayments.EOF
```

```
            clRowCount = clRowCount + 1    'increment
```

```
            With Grid
```

If .Rows = 2 And .TextMatrix(1, 1) = "" Then

.TextMatrix(1, 1) = rsPayments!RecievedPaymentID

.TextMatrix(1, 2) = rsPayments!Date

.TextMatrix(1, 3) = rsPayments!PaymentTypeID

.TextMatrix(1, 4) = rsPayments!PaymentType

.TextMatrix(1, 5) = toMoney(rsPayments!Amount)

Else

.Rows = .Rows + 1

.TextMatrix(.Rows - 1, 1) = rsPayments!RecievedPaymentID

.TextMatrix(.Rows - 1, 2) = rsPayments!Date

.TextMatrix(.Rows - 1, 3) = rsPayments!PaymentTypeID

.TextMatrix(.Rows - 1, 4) = rsPayments!PaymentType

.TextMatrix(.Rows - 1, 5) = toMoney(rsPayments!Amount)

End If

End With

' AmountPaid = rsPayments!Amount

rsPayments.MoveNext

Wend

Grid.Row = 1

Grid.ColSel = 5

```
'Set fixed cols
```

```
Grid.FixedRows = Grid.Row: 'Grid.SelectionMode = flexSelectionFree
```

```
Grid.FixedCols = 1
```

```
End If
```

```
rsPayments.Close
```

```
'Clear variables
```

```
Set rsPayments = Nothing
```

```
Exit Sub
```

```
err:
```

```
If err.Number = 94 Then Resume Next
```

```
prompt_err err, Name, "DisplayForEditing"
```

```
Screen.MousePointer = vbDefault
```

```
End Sub
```

```
Private Sub DeleteItems()
```

```
Dim CurrRow As Integer
```

```
Dim rsPayments As New Recordset
```

```

rsPayments.CursorLocation = adUseClient

rsPayments.Open "SELECT * FROM [Payments Received] WHERE AccRecID=" & PK, CN,
adOpenStatic, adLockOptimistic

If rsPayments.RecordCount > 0 Then

    rsPayments.MoveFirst

    While Not rsPayments.EOF

        CurrRow = getFlexPos(Grid, 1, rsPayments!RecievedPaymentID)

        'Add to grid

        With Grid

            If CurrRow < 0 Then

                'Delete record if doesnt exist in flexgrid

                DelRecwSQL "Payments", "RecievedPaymentID", "", True,
rsPayments!RecievedPaymentID

            End If

        End With

        rsPayments.MoveNext

    Wend

End If

End Sub

Private Sub txtAmount_GotFocus()

    HLText txtAmount

```


End Sub

Private Sub txtAmount_KeyPress(KeyAscii As Integer)

 KeyAscii = isNumber(KeyAscii)

End Sub

Private Sub txtAmount_Validate(Cancel As Boolean)

 txtAmount.Text = toMoney(txtAmount.Text)

End Sub

FrmCheckInList(code)

Public PK As Long

Public Company As String

Public Balance As Currency

Public AmountPaid As Currency

Dim cIRowCount As Integer

Dim Amount As Currency

Private Sub btnRemove_Click()

 'Remove selected load product

 With Grid

 'Update the record count

 cIRowCount = cIRowCount - 1

 If .Rows = 2 Then Grid.Rows = Grid.Rows + 1

 .RemoveItem (.RowSel)

 End With

 btnRemove.Visible = False

```
txtBalance.Text = toMoney(toNumber(txtBalance.Text) + toNumber(Grid.TextMatrix(Grid.RowSel, 5)))
```

```
txtAmountPaid.Text = toMoney(txtAmountPaid.Text) - toNumber(Grid.TextMatrix(Grid.RowSel, 5))
```

```
Grid_Click
```

```
End Sub
```

```
Private Sub cmdAdd_Click()
```

```
    If Trim(txtAmount.Text) = "0.00" Then Exit Sub
```

```
    If toNumber(txtAmount) > Balance Then
```

```
        MsgBox "Amount paid exceed balance. Please enter correct amount.", vbInformation
```

```
        txtAmount.SetFocus
```

```
    Exit Sub
```

```
End If
```

```
Dim CurrRow As Integer
```

```
'Add to grid
```

```
With Grid
```

```
    'Perform if the record is not exist
```

If .Rows = 2 And .TextMatrix(1, 2) = "" Then

.TextMatrix(1, 2) = dtpDate.Value

.TextMatrix(1, 3) = dcPaymentType.BoundText

.TextMatrix(1, 4) = dcPaymentType.Text

.TextMatrix(1, 5) = toMoney(txtAmount.Text)

Else

.Rows = .Rows + 1

.TextMatrix(.Rows - 1, 2) = dtpDate.Value

.TextMatrix(.Rows - 1, 3) = dcPaymentType.BoundText

.TextMatrix(.Rows - 1, 4) = dcPaymentType.Text

.TextMatrix(.Rows - 1, 5) = toMoney(txtAmount.Text)

.Row = .Rows - 1

End If

'Increase the record count

cIRowCount = cIRowCount + 1

txtBalance.Text = toMoney(Balance - toNumber(txtAmount.Text))

' Balance = Balance - toNumber(txtAmountPaid.Text)

txtAmountPaid.Text = toMoney(toNumber(txtAmountPaid.Text) + toNumber(txtAmount.Text))

'Highlight the current row's column

.ColSel = 5

'Display a remove button

Call Grid_Click

Call ResetFields

End With

End Sub

Private Sub cmdCancel_Click()

Unload Me

End Sub

Public Sub cmdSave_Click()

Dim rsPayments As New Recordset

rsPayments.CursorLocation = adUseClient

rsPayments.Open "SELECT * FROM [Accounts Receivable] WHERE AccRecID=" & PK, CN,
adOpenStatic, adLockOptimistic

With rsPayments

.Fields("Debit") = txtAmountPaid.Text

.Update

End With

DeleteItems

Dim rsPaymentRec As New Recordset

rsPaymentRec.CursorLocation = adUseClient

rsPaymentRec.Open "SELECT * FROM [Payments Received] WHERE AccRecID=" & PK, CN,
adOpenStatic, adLockOptimistic

Dim c As Integer

With Grid

'Save the details of the records

For c = 1 To cIRowCount

.Row = c

If .TextMatrix(c, 1) = "" Then

rsPaymentRec.AddNew

rsPaymentRec![AccRecID] = PK

Else

rsPaymentRec.Filter = "RecievedPaymentID = " & toNumber(.TextMatrix(c, 1))

```

'      If rsPaymentRec.RecordCount = 0 Then
'
'          rsPaymentRec.AddNew
'
'
'          rsPaymentRec![AccRecID] = PK
'
'      End If
'
'      End If

      rsPaymentRec![Date] = .TextMatrix(c, 2)

      rsPaymentRec![PaymentTypeID] = .TextMatrix(c, 3)

      rsPaymentRec![Amount] = .TextMatrix(c, 5)


      rsPaymentRec.Update

  Next c

End With

'Clear variables

c = 0

Set rsPayments = Nothing

Set rsPaymentRec = Nothing


Unload frmAccountReceivable

```

End Sub

Private Sub cmdUpdate_Click()

If Trim(txtAmount.Text) = "0.00" Then Exit Sub

txtBalance.Text = toMoney(txtBalance.Text) + Amount

txtAmountPaid.Text = toMoney(txtAmountPaid.Text) - Amount

txtBalance.Text = toMoney(toNumber(txtBalance.Text) - toNumber(txtAmount.Text))

txtAmountPaid.Text = toMoney(toNumber(txtAmountPaid.Text) + toNumber(txtAmount.Text))

With Grid

.TextMatrix(.RowSel, 2) = dtpDate.Value

.TextMatrix(.RowSel, 3) = dcPaymentType.BoundText

.TextMatrix(.RowSel, 4) = dcPaymentType.Text

.TextMatrix(.RowSel, 5) = toMoney(txtAmount.Text)

End With

Call Grid_Click

Call ResetFields

End Sub


```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        SendKeys "{TAB}"
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Call InitGrid
```

```
    bind_dc "SELECT * FROM [Payment Type]", "PaymentType", dcPaymentType, "paymentTypeID",  
    True
```

```
    txtCompany.Text = Company
```

```
    txtBalance.Text = toMoney(Balance)
```

```
    txtAmountPaid.Text = toMoney(AmountPaid)
```

```
    dtpDate.Value = Date
```

```
    DisplayForEditing
```

```
    If Balance = "0.00" Then
```

```
        Grid.Height = 3000
```

Grid.Top = 2160

dtpDate.Visible = False

dcPaymentType.Visible = False

txtAmount.Visible = False

cmdAdd.Visible = False

cmdUpdate.Visible = False

lblPaid.Visible = True

End If

End Sub

'Procedure used to initialize the grid

Private Sub InitGrid()

 cIRowCount = 0

 With Grid

 .Clear

 .ClearStructure

 .Rows = 2

 .FixedRows = 1

 .FixedCols = 1

 .Cols = 6

 .ColSel = 5

'Initialize the column size

.ColWidth(0) = 315

.ColWidth(1) = 0

.ColWidth(2) = 1200

.ColWidth(3) = 0

.ColWidth(4) = 1200

.ColWidth(5) = 1200

'Initialize the column name

.TextMatrix(0, 0) = ""

.TextMatrix(0, 1) = "RecievedPaymentID"

.TextMatrix(0, 2) = "Date"

.TextMatrix(0, 3) = "Payment Type ID"

.TextMatrix(0, 4) = "Payment Type"

.TextMatrix(0, 5) = "Amount"

End With

End Sub

Private Sub Form_Unload(Cancel As Integer)

frmAccountReceivableList.RefreshRecords

Set frmAccountReceivable = Nothing

End Sub

Private Sub Grid_Click()

With Grid

 dtpDate.Value = .TextMatrix(.RowSel, 2)

 dcPaymentType.BoundText = .TextMatrix(.RowSel, 3)

 txtAmount.Text = toMoney(.TextMatrix(.RowSel, 5))

 Amount = .TextMatrix(.RowSel, 5)

 If Grid.Rows = 2 And Grid.TextMatrix(1, 2) = "" Then

 btnRemove.Visible = False

 Else

 btnRemove.Visible = True

 btnRemove.Top = (Grid.CellTop + Grid.Top) - 20

 btnRemove.Left = Grid.Left + 50

 End If

End With

End Sub

Private Sub ResetFields()

 txtAmount.Text = ""

```

txtAmount.SetFocus

End Sub

Private Sub DisplayForEditing()

    On Error GoTo err

    'Display the details

    Dim rsPayments As New Recordset

    cIRowCount = 0

    rsPayments.CursorLocation = adUseClient

    rsPayments.Open "SELECT * FROM [qry_Payment_Received] WHERE AccRecID=" & PK, CN,
adOpenStatic, adLockOptimistic

    If rsPayments.RecordCount > 0 Then

        rsPayments.MoveFirst

        While Not rsPayments.EOF

            cIRowCount = cIRowCount + 1    'increment

            With Grid

                If .Rows = 2 And .TextMatrix(1, 1) = "" Then

                    .TextMatrix(1, 1) = rsPayments!RecievedPaymentID

```

```
.TextMatrix(1, 2) = rsPayments!Date  
  
.TextMatrix(1, 3) = rsPayments!PaymentTypeID  
  
.TextMatrix(1, 4) = rsPayments!PaymentType  
  
.TextMatrix(1, 5) = toMoney(rsPayments!Amount)
```

Else

```
.Rows = .Rows + 1  
  
.TextMatrix(.Rows - 1, 1) = rsPayments!RecievedPaymentID  
  
.TextMatrix(.Rows - 1, 2) = rsPayments!Date  
  
.TextMatrix(.Rows - 1, 3) = rsPayments!PaymentTypeID  
  
.TextMatrix(.Rows - 1, 4) = rsPayments!PaymentType  
  
.TextMatrix(.Rows - 1, 5) = toMoney(rsPayments!Amount)
```

End If

End With

```
' AmountPaid = rsPayments!Amount
```

```
rsPayments.MoveNext
```

Wend

```
Grid.Row = 1
```

```
Grid.ColSel = 5
```

```
'Set fixed cols
```

```
Grid.FixedRows = Grid.Row: 'Grid.SelectionMode = flexSelectionFree
```

```
Grid.FixedCols = 1
```

```
End If
```

```
rsPayments.Close
```

```
'Clear variables
```

```
Set rsPayments = Nothing
```

```
Exit Sub
```

```
err:
```

```
If err.Number = 94 Then Resume Next
```

```
prompt_err err, Name, "DisplayForEditing"
```

```
Screen.MousePointer = vbDefault
```

```
End Sub
```

```
Private Sub DeleteItems()
```

```
Dim CurrRow As Integer
```

```
Dim rsPayments As New Recordset
```

```
rsPayments.CursorLocation = adUseClient
```

```
rsPayments.Open "SELECT * FROM [Payments Received] WHERE AccRecID=" & PK, CN,  
adOpenStatic, adLockOptimistic
```

```

If rsPayments.RecordCount > 0 Then

    rsPayments.MoveFirst

    While Not rsPayments.EOF

        CurrRow = getFlexPos(Grid, 1, rsPayments!RecievedPaymentID)

        'Add to grid

        With Grid

            If CurrRow < 0 Then

                'Delete record if doesnt exist in flexgrid

                DelRecwSQL "Payments", "RecievedPaymentID", "", True,
rsPayments!RecievedPaymentID

                End If

            End With

            rsPayments.MoveNext

        Wend

    End If

End Sub

```

```

Private Sub txtAmount_GotFocus()

    HLText txtAmount

End Sub

```

```

Private Sub txtAmount_KeyPress(KeyAscii As Integer)

```



```
KeyAscii = isNumber(KeyAscii)
```

```
End Sub
```

```
Private Sub txtAmount_Validate(Cancel As Boolean)
```

```
txtAmount.Text = toMoney(txtAmount.Text)
```

```
End Sub
```

```
FrmCheckOut(code)
```

```
Public RoomNumber As Integer
```

```
Public AmountPaid As Currency 'Amount paid from frmPayment
```

```
Public OtherCharges As Currency
```

```
Public AutoCheckOut As Boolean
```

```
Dim RS As New Recordset
```

```
Private Sub cmdCancel_Click()
```

```
On Error GoTo err
```

```
CN.BeginTrans
```

```
CN.Execute "DELETE FolioNumber " & _
```

"From [Rate Per Period] " & _

"WHERE FolioNumber=" & txtGuestName.Tag & ""

CN.Execute "INSERT INTO [Rate Per Period] " & _

"SELECT [Rate Per Period Temp].* " & _

"FROM [Rate Per Period Temp] " & _

"Where ((([Rate Per Period Temp].FolioNumber) = " & txtGuestName.Tag & "")) " & _

"ORDER BY [Rate Per Period Temp].Date;"

CN.CommitTrans

Unload Me

Exit Sub

err:

CN.RollbackTrans

prompt_err err, Name, "cmdCancel_Click"

Screen.MousePointer = vbDefault

End Sub

Private Sub cmdCheckOut_Click()

On Error GoTo err

If txtBalance.Text <> "0.00" Then

MsgBox "There's still remaining balance for this guest.", vbExclamation

Exit Sub

End If

If MsgBox("Are you sure you want to Check Out?", vbYesNo + vbInformation) = vbNo Then Exit Sub

CN.BeginTrans

ChangeValue CN, "Rooms", "RoomStatusID", 3, True, "WHERE RoomNumber = " &
txtRoomNumber.Text

Call frmPayment.cmdSave_Click

Call frmOtherCharges.cmdSave_Click

With RS

'Delete record from Inventory

CN.Execute "DELETE ID, Status " & _

"From [Inventory] " & _

"WHERE ID=" & .Fields("FolioNumber") & " AND Status='Check In'"

```
.Fields("DateOut") = dtpDateOut.Value
```

```
.Fields("OtherCharges") = txtOtherCharges.Text
```

```
.Fields("Discount") = txtDiscount.Text
```

```
.Fields("AmountPaid") = txtAmountPaid.Text
```

```
.Fields("Days") = txtDays.Text
```

```
.Fields("Status") = "Check Out"
```

```
.Fields("CheckOutBy") = CurrUser.USER_PK
```

```
.Update
```

```
End With
```

```
CN.CommitTrans
```

```
Call PrintFolio
```

```
RS.Close
```

```
Set RS = Nothing
```

```
Unload Me
```

```
Exit Sub
```

err:

CN.RollbackTrans

prompt_err err, Name, "cmdCheckOut_Click"

Screen.MousePointer = vbDefault

End Sub

Private Sub PrintFolio()

With frmReports

.strReport = "Folio"

.strWhere = "{qry_RPT_Customers.FolioNumber} = " & txtGuestName.Tag & " AND
{qry_RPT_Customers.Status} = 'Check Out'"

frmReports.Show vbModal

End With

End Sub

Private Sub dtpDateOut_Change()

txtDays.Text = dtpDateOut.Value - CDate(txtDateIn.Text)

Call ComputeRate

End Sub

```
Private Sub dtpDateOut_LostFocus()
```

```
    If CDate(txtDateIn.Text) > dtpDateOut.Value Then
```

```
        MsgBox "Check In date must be below check out date. Please enter another check out date.",  
vbInformation
```

```
        dtpDateOut.SetFocus
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        SendKeys "{TAB}"
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    On Error GoTo err
```

```
    CN.BeginTrans
```

```
    RS.CursorLocation = adUseClient
```

```
    RS.Open "SELECT * FROM Transactions WHERE RoomNumber = " & RoomNumber & " AND  
Status = 'Check In'", CN, adOpenStatic, adLockOptimistic
```

```
bind_dc "SELECT * FROM [Rate Type]", "RateType", dcRateType, "RateTypeID", True
```

```
txtRoomNumber.Text = RoomNumber
```

```
With RS
```

```
    txtGuestName.Tag = .Fields("FolioNumber")
```

```
    txtGuestName.Text = getValueAt("SELECT [Name] FROM qry_CheckIn WHERE FolioNumber =  
    "" & .Fields("FolioNumber") & " ", "Name")
```

```
    txtDateIn.Text = .Fields("DateIn")
```

```
    If AutoCheckOut = True Then
```

```
        If .Fields("DateOut") >= Date Then
```

```
            dtpDateOut.Value = .Fields("DateOut")
```

```
        Else
```

```
            dtpDateOut.Value = Date
```

```
        End If
```

```
    Else
```

```
        dtpDateOut.Value = .Fields("DateOut")
```

```
    End If
```

```
    dcRateType.BindText = .Fields("RateType")
```

```
    txtDays.Text = dtpDateOut.Value - CDate(txtDateIn.Text)
```

```
    txtAdults.Text = .Fields("Adults")
```

```
    txtChildrens.Text = .Fields("Childrens")
```

```
txtRate.Text = toMoney(.Fields("Rate"))
```

```
txtOtherCharges.Text = toMoney(.Fields("OtherCharges"))
```

```
txtDiscount.Text = toMoney(.Fields("Discount"))
```

```
txtAmountPaid.Text = toMoney(.Fields("AmountPaid"))
```

End With

```
dcRateType.Enabled = False
```

Call ComputeAddRate

Call ComputeRate

```
CN.Execute "DELETE FolioNumber " & _
```

```
"From [Rate Per Period Temp] " & _
```

```
"WHERE FolioNumber=" & txtGuestName.Tag & ""
```

```
CN.Execute "INSERT INTO [Rate Per Period Temp] " & _
```

```
"SELECT [Rate Per Period].* " & _
```

```
"From [Rate Per Period] " & _
```

```
"WHERE FolioNumber=" & txtGuestName.Tag & ""
```

```
CN.CommitTrans
```


Exit Sub

err:

CN.RollbackTrans

prompt_err err, Name, "txtDays_Change"

Screen.MousePointer = vbDefault

End Sub

Private Sub ComputeRate()

txtTotalCharges.Text = toMoney(ComputeRatePerPeriod)

txtSubTotal.Text = toMoney(toNumber(txtTotalCharges.Text) + toNumber(txtOtherCharges.Text))

txtTotal.Text = toMoney(toNumber(txtSubTotal.Text) - (toNumber(txtSubTotal.Text) *
toNumber(txtDiscount.Text) / 100))

txtBalance.Text = toMoney(toNumber(txtTotal.Text) - toNumber(txtAmountPaid.Text))

End Sub

'Compute additional rate (no. of days & childrens)

Private Sub ComputeAddRate()

Dim rsRoomRates As New ADODB.Recordset

With rsRoomRates

```
.Open "SELECT * FROM [Room Rates] WHERE RoomNumber = " & RoomNumber & " AND  
RateTypeID = " & dcRateType.BindText, CN, adOpenStatic, adLockOptimistic
```

```
If .RecordCount > 0 Then
```

```
    txtRate.Text = toMoney(!RoomRate)
```

```
    txtAdults.Tag = !ExtraAdultRate
```

```
    txtChildrens.Tag = !ExtraChildRate
```

```
End If
```

```
End With
```

```
rsRoomRates.Close
```

```
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    lblRatePerPeriod.FontUnderline = False
```

```
    lblAmountPaid.FontUnderline = False
```

```
    lblOtherCharges.FontUnderline = False
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    frmRoomsWindow.RefreshRecords
```

```
    Unload frmPayment
```

```

Set frmPayment = Nothing

Set frmCheckOut = Nothing

End Sub

Private Sub lblAmountPaid_Click()

    With frmPayment

        .FolioNumber = txtGuestName.Tag

        .GuestName = txtGuestName.Text

        .Balance = txtBalance.Text

        .RefreshBalance

        Set .RefForm = Me

        .Show vbModal

        txtAmountPaid.Text = toMoney(AmountPaid)

    End With

End Sub

Private Sub lblAmountPaid_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

    SetHandCur True

    lblAmountPaid.FontUnderline = True

End Sub

```

```
Private Function ComputeRatePerPeriod() As Currency
```

```
    Dim rsRoomRates As New ADODB.Recordset
```

```
    With rsRoomRates
```

```
        .Open "SELECT * FROM [Rate Per Period] WHERE FolioNumber = " & txtGuestName.Tag & "",  
        CN, adOpenStatic, adLockOptimistic
```

```
        Do Until .EOF
```

```
            ComputeRatePerPeriod = ComputeRatePerPeriod + toMoney(!Rate) + toMoney(!Adults) +  
            toMoney(!Childrens)
```

```
        .MoveNext
```

```
    Loop
```

```
End With
```

```
rsRoomRates.Close
```

```
End Function
```

```
Private Sub lblOtherCharges_Click()
```

```
    With frmOtherCharges
```

```
        .FolioNumber = txtGuestName.Tag
```

```
        .GuestName = txtGuestName.Text
```

```
Set .RefForm = Me
```

```
.Show vbModal
```

```
txtOtherCharges.Text = toMoney(OtherCharges)
```

```
End With
```

```
End Sub
```

```
Private Sub lblOtherCharges__MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
SetHandCur True
```

```
lblOtherCharges.FontUnderline = True
```

```
End Sub
```

```
Private Sub lblRatePerPeriod_Click()
```

```
With frmRatePerPeriod
```

```
.FolioNumber = txtGuestName.Tag
```

```
.Show vbModal
```

```
Call ComputeRate
```

```
End With
```

```
End Sub
```

```
Private Sub lblRatePerPeriod_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    SetHandCur True
```

```
    lblRatePerPeriod.FontUnderline = True
```

```
End Sub
```

```
Private Sub txtAmountPaid_Change()
```

```
    txtBalance.Text = toMoney(toNumber(txtTotal.Text) - toNumber(txtAmountPaid.Text))
```

```
End Sub
```

```
Private Sub txtDays_Change()
```

```
On Error GoTo err
```

```
    Dim rsRatePerPeriod As New ADODB.Recordset
```

```
    Dim tmpDate As Date
```

```
    Dim minNoofPerson As Integer
```

```
    tmpDate = txtDateIn.Text
```

```
    If txtAdults.Tag = "" Then Exit Sub
```

```
    CN.BeginTrans
```

CN.Execute "DELETE [Date] " & _

"FROM [Rate Per Period] " & _

"WHERE [Date]>#" & dtpDateOut - 1 & "#"

Dim intAdults As Integer

minNoofPerson = getValueAt("SELECT * FROM [Room Rates] WHERE RoomNumber = " &
RoomNumber & " AND RateTypeID = " & dcRateType.BoundText, "NoofPerson")

If txtAdults.Text = minNoofPerson Then

intAdults = 0

Else

intAdults = CInt(txtAdults.Text) - minNoofPerson

End If

With rsRatePerPeriod

.Open "SELECT * FROM [Rate Per Period] WHERE FolioNumber = " & txtGuestName.Tag & "
ORDER BY [Date]", CN, adOpenStatic, adLockOptimistic

Do Until tmpDate > dtpDateOut.Value - 1

.Filter = "[Date] = #" & tmpDate & "#"

```

If .RecordCount = 0 Then

    .AddNew

    .Fields("FolioNumber") = txtGuestName.Tag

    .Fields("Date") = tmpDate

    .Fields("RoomNumber") = txtRoomNumber.Text

    .Fields("RateTypeID") = dcRateType.BoundText

    .Fields("Rate") = txtRate.Text

    .Fields("Adults") = txtAdults.Tag * intAdults

    .Fields("Childrens") = toMoney(txtChildrens.Tag) * toNumber(txtChildrens.Text)

    .Update

End If

tmpDate = tmpDate + 1

Loop

End With

CN.CommitTrans

rsRatePerPeriod.Close

Exit Sub

```


Appendix 2: Questionnaire

Staff Questionnaire

a) Bio-Data

Names.....

Age.....

Sex.....

Nationality.....

b) Education Background

(Tick where necessary)

O level ☐

A level ☐

University ☐

Computer Literacy

Yes ☐

No ☐

c) Department *(Tick where necessary)*

Accounts ☐

Reception ☐

Management ☐

Others Specify ☐

Client Questionnaire

a) Guest Information

Name.....

Address.....

Country.....

Tel. Number.....

Nationality.....

b) Identification Information

ID Type.....

ID Number.....

c) Rate Information

Date in

Date out.....

Room

Single Double

Room Number.....

Number of Adults

Number of Children.....

Number of days.....

Rate / Period.....

Charges.....

Total Charges.....