



The Apartment Rental System

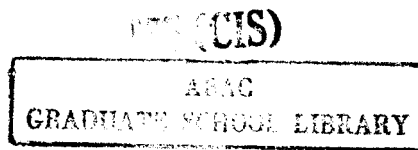
By

Mr. Chusak Sirayanyong

Final Report of the Three - Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

July, 1998



THE APARTMENT RENTAL SYSTEM

By

Mr.Chusak Sirayanyong

Final Report of the Three-Credit Course
CS 6998 System Development Project

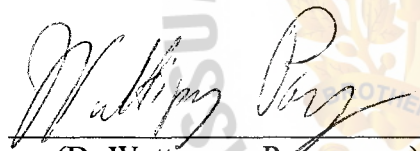
Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

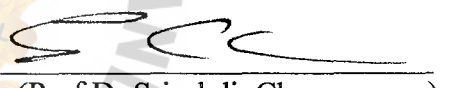
July, 1998

Project Title : The Apartment Rental System
Name : Mr.Chusak Sirayanyong
Project Advisor : Dr.Wuttipong Pongsuwan
Academic Year : 1998

The Graduate School of Assumption University had approved this final report of the three-credit course, CS 6998 System Development Project, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems.

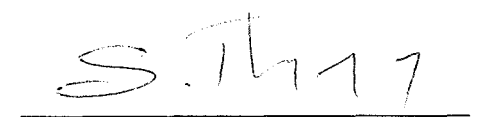
Approval Committee :


(Dr. Wuttipong Pongsuwan)
Advisor


(Prof. Dr. Srisakdi Charmonman)
Chairman


(Air Marshal Dr. Chulit Meesajjee)
Dean and Co-advisor


(Dr. Sudhiporn Patumtaewapibal)
Member


(Assoc. Prof. Somchai Thayarnyong)
MUA Representative

July 1998

Abstract

The system development of this project includes the analysis, design and implementation plan of a computerized system for apartment front office operation and management. The system has been designed to cover the function of reservation, guest registration (check-in), check out, meal, laundry, payment, extended stay, assigning rooms and rooms' management.

Scenario analysis and design were used to analyze the system requirement. The data processed through the organization are represented using object orientation method. The Entity Relationship diagram is used to represent the data definition and their relationship.

The project used the web browser (VbScript and JavaScript) as the front end to access the distributed database system by perl programming language to execute the variable from html document and retrieve the wanted data by SQL statement and sent back to generate document page as the result. Every O/S and every browser program can use this system. It is not limited by the software or the hardware.

Acknowledgements

The author would like to acknowledge the contributions of the following organization, without whom the completion of this project would not be possible Nippon Electric Corporation Japan for providing the document and information of system requirements on database distribution and perl programming functions. I also would like to convey my sincere gratitude to Dr.Wuttipong Pongsuwan, a project advisor who provided advice and comments throughout the development of this project.

Lastly, I would like to extend my appreciation to all my teachers in Computer Information System (CIS) department of Assumption University for providing knowledge and education throughout my academic years in university.

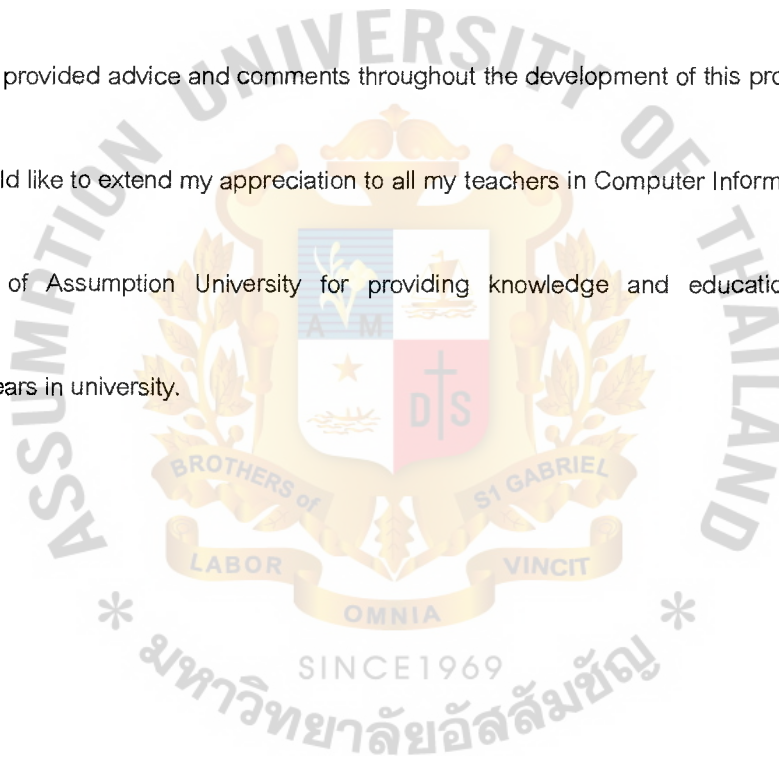


Table of Contents

1. INTRODUCTION	4
1.1 BACKGROUND OF PROJECT	4
1.2 OBJECTIVES OF THE PROJECT	6
1.3 SCOPE OF THE PROJECT	6
2. EXISTING SYSTEM	8
2.1 GENERAL APARTMENT ORGANIZATION	8
2.2 CURRENT PROBLEMS AND AREAS OF IMPROVEMENTS	13
3. PROPOSED SYSTEM	15
3.1 USER REQUIREMENT	15
3.2 SYSTEMS DESIGN	21
3.2.2 Scenario Analysis	30
3.2.3 Screen Design	35
3.3 HARDWARE AND SOFTWARE REQUIREMENT	45
3.4 SECURITY AND CONTROLS	47
3.5 COST / BENEFIT ANALYSIS	50
4. PROJECT IMPLEMENTATION	55
4.1 IMPLEMENTATION PROCEDURES	55
5. CONCLUSIONS AND RECOMMENDATION	61
6. REFERENCES	63
APPENDIX A	64
CLASS AND FUNCTION DESIGN	64
DATABASE DESIGN	83
APPENDIX B	87
SCREEN CODE	87
SERVER CODE(PERL SCRIPT)	147

LIST OF FIGURES

Figure 3.1 APARTMENT RENTAL SYSTEM OBJECTS MODEL DIAGRAM	21
Figure 3.2 RESERVATION	21
Figure 3.3 CHECK IN	22
Figure 3.4 ASSIGN ROOM	22
Figure 3.5 EXTENDED STAY	23
Figure 3.6 MEAL AND LAUNDRY	23
Figure 3.7 CHECK OUT	24
Figure 3.8 PAYMENT	24
Figure 3.9 RESERVATION SCENARIO	25
Figure 3.10 ASSIGN SCENARIO	26
Figure 3.11 CHECK IN SCENARIO	27
Figure 3.12 EXTENDED SCENARIO	28
Figure 3.13 CHECK OUT SCENARIO	29
Figure 3.14 TABLE OF CONTENTS (START PAGE)	35
Figure 3.15 RESERVATION	35
Figure 3.16 RESERVATION_OK	36
Figure 3.17 RESERVATION_ERROR	36
Figure 3.18 CANCEL RESERVATION	37
Figure 3.19 GET RESERVATION BY DATE	37
Figure 3.20 CHECK OUT REPORT	38
Figure 3.21 CHECK OUT DETAIL	38
Figure 3.22 CHECK OUT GETS ROOM NUMBER	39
Figure 3.23 REGISTRATION REPORT	39
Figure 3.24 EXTENDED ROOM UPDATE OK	40
Figure 3.25 REPORT GETS RESERVE ID FROM DATE	40
Figure 3.26 LAUNDRY	42
Figure 3.27 LAUNDRY UPDATE OK	42
Figure 3.28 LAUNDRY UPDATE ERROR	43
Figure 3.29 REGISTRATION GETS CUSTOMER INFORMATION	43
Figure 3.30 REGISTER UPDATE OK	44
Figure 3.31 REGISTER REQUEST RESERVATION ID	44

LIST OF APPENDIX TABLES

Table A-1 CLASS BILL	63
Table A-2 CLASS CARDS	64
Table A-3 CLASS CLOTHES	65
Table A-4 CLASS CORPORATE	66
Table A-5 CLASS LAUNDRY	67
Table A-6 CLASS MEAL	68
Table A-7 CLASS PAYMENT	69
Table A-10 CLASS REGISTRATION	70
Table A-11 CLASS RESERVE_DETAIL	71
Table A-12 CLASS RESERVATION	72
Table A-13 CLASS RESERVATION DETAIL	73
Table A-14 FUNCTION RET_TOT_MEAL_LAUNDRY_AMT	74
Table A-15 FUNCTION CLOSE_BILL	74
Table A-16 FUNCTION RET_CLOTH_PRICE	75
Table A-17 FUNCTION CAL_LAUN_AMT	76
Table A-18 FUNCTION UPDATE_MEAL_AMT	76
Table A-19 FUNCTION CAL_TOT_AMT	77
Table A-20 FUNCTION RET_PAY_TYPE_LIST	78
Table A-21 FUNCTION CAL_NET_TOT_AMT	78
Table A-22 FUNCTION RET_ACT_CHECK_OUT_LIST	79
Table A-23 FUNCTION ASSIGN_ROOM_NO	80
Table A-24FUNCTION RET_RESERVED_ROOM	81
Table A-25 FUNCTION UPDATE_REGISTRATION	81
Table A-26:RESERVATION TABLE	82
Table A-27 RESERVE_DETAIL TABLE	82
Table A-28 REGISTRATION TABLE	83
Table A-29 LAUNDRY TABLE	83
Table A-30 COTHES TABLE	83
Table A-31 PAYMENT TABLE	84
Table A-32 ROOMS TABLE	84
Table A-33 BILL TABLE	84
Table A-34 MEAL TABLE	85
Table A-35 CORPORATE TABLE	85
Table A-36 CARDS TABLE	85

1. Introduction

1.1 Background of project

Apartment for rent is a good business in 1996. Those businesses provide services, primarily accommodation food and beverages, not only for those travelling for pleasure but also for the business travelers.

Regardless of how the apartment is organized, front office is always an essential focal point or the nerve center of any apartment operation. Being the nerve center of a modern apartment, front office must be aware of what is happening at virtually every level of apartment organization structure. One of the negative aspects is that the front office also serves as a point for guest complaints. One of the major sources of complaints that is related to the front office do as have to do with the accuracy of guest reservations and details of attendant.

Controlling accurate reservations record on any scale is extremely difficult. It is necessary to automate the reservation process, so that the reservation can be managed and controlled effectively. The automated system will ensure that all the important information is not missing. Also, a high level of accuracy of reservation records is maintained.

The front office is not only responsible for the reservation and registration. Other functions that are responsible by the front office include providing information on services, coordination of activities, rooms status reports, maintenance of guest accounts and settlement and collection of guest accounts.

Computer-based information system greatly reduces the difficulty of controlling and operating tasks. A well-design computer information system can handle all the massive paperwork, repetitive and tedious tasks and the pastidious task of room assignment.

The system presented in this project is to serve the common function of the front counter. The information proceeded in those activities are almost the same for most small to medium sized apartments.

1.2 Objectives of the project

Following are the broad objectives of the front office information system:

1. To determine and analyze business and users' requirements in front office apartment management system;
2. To design an information system to assist the front office operation;
3. To develop an effective low cost application and low cost hardware;
4. To estimate and compare the cost and the benefit between the manual and the computer system;
5. To perform the system testing on the complete software system.

1.3 Scope of the project

The project covers the major parts of the front office hotel management and operation. The following subsystems will be included in the project: -

1. *The sales of rooms*

- Accept reservation
- Handle walk in customers
- Perform registration procedure
- Assign the room

- Extend the stay

2. Room management

- Room status
- Guest information
- Coordination of room status and house keeping

3. Guest accounting

- Creation of account
- Meal charge
- Entry transaction charge
- Auditing procedures
- Settlement of account
- Laundry charge
- Last minute charge
- Guest check out



2. Existing System

Since the system development presented in this project is not designed for a specific apartment, the information in this section does not describe the characteristics of any organization. Rather, this section describes the characteristics of apartment organization in general. Existing business functions are common functions found in almost all small to medium sized apartments. Current problem and areas of improvements are defined with assumption that there is not existing computerized system utilized for apartment's front office operations and management.

2.1 General apartment organization

The organization of an apartment depends on the size of its staff. The number of employees that are required to operate the apartment efficiently varies with the numbers of guests. In turn, the number of rooms limits the number of guest. A middle sized apartment also provides food and beverage services and maintains full time staffs. A typical middle sized apartment is divided into eight basic functions, as follows:

1. Administration
2. Rooms

3. Food and beverage service
4. Guest services
5. Marketing and sales
6. Accounting
7. Security
8. Engineering

The front office

The front office staff is the public's main contact with the apartment. The staff members handle reservations, greet guests on arrival, register new guests, dispense keys, handle incoming and outgoing mail, take messages for guests, provide information, listen to complaints, and handle check-out procedures when guests depart. The following personnel may be employed in the front office:

- Front office manager
- Assistant front office manager
- Front desk representatives
- Night Auditor
- Cashiers
- Reservation

- Operator

Besides public contact and guest communications, the front office staff may also be required to organize room occupancy and reservation statistics, maintain ongoing communication with the housekeeping department, promote room sales and balance the books.

At most apartments, the front desk remains active 24 hours a day. Therefore, three shifts are required to staff the front office. The following shifts are common:



Morning day shift	7 A.M. – 3 P.M.
Day evening shift	3 P.M. – 11 P.M.
Night shift	11 P.M. – 7 A.M.

Front desk representatives

When guests arrive at the hotel, the front desk representatives is responsible for verifying reservation information, checking credit identification and authorization, assigning rooms, and dispensing room keys.

When a guest is ready to depart, the front office representative must prepare, verify, and arrange the guest check. Other duties may include checking the room rack at the beginning of the shift, reviewing

reservations for the current date, and communicating with housekeeping department regarding occupancy forecasts and room needs.

The night auditor

A night auditor must perform the duties of a front desk representative at night and the bookkeeping function i.e., prepare the machine balance report. When front desk activity slackens, the night auditor begins to audit, or trace the posting of the previous day's transactions to verify their accuracy. The night auditor calculates the total charges owed by the guest during the day. These are subtracted from the total charge to determine the daily balance. The balance represents the amount that is still owed to the apartment for the previous day's transactions.

Cashiers

A front desk cashier is responsible for checking out departing guests, posting charges, verifying the guest check, and handling payments. The cashier may be responsible for calculating the charges and presenting the guest check.

The reservation operations

Reservation operators are responsible for handling advance room requests, determining availability, quoting rates, and documenting reservations.

The telephone operator

An operator answers all incoming telephone call and directs callers to the appropriate guestroom. The

operator must have an up to date directory of guests' room assignments.

2.2 Current problems and areas of improvements

The clerical staff member who performs the reservation function handles a number of tasks that are important to the orderly accumulation and dissemination of reservation information to the appropriate department. In a reservation process, the following difficulties arise:

- difficulties handling reservation years in advance
- extremely difficulties in controlling accuracy of reservation records
- a lot of paper work involved
- difficulties in determining room availability during high season
- making a reservation is time consuming and prone to error

The most highly visible duty of the front office is to register and checkout guests. Check-in and check out procedures consist of several steps, each of which is important to efficient room management and

the maintenance of a pleasant and orderly atmosphere at the front office. The following are the

problems normally found at the front office are as follows:

- preparing arrival list is very time consuming
- guests have to wait for quite a while when the front office is busy
- error in reservation can make guest registration process difficult
- room assignment task is tedious in order to select the most appropriate room
- room status determination can be quite confusing to apartment staffs
- repletion task of guest account balance update is required every day
- preparation of guest departure list is time consuming
- preparing summary of accounting information is difficult
- extremely difficult to prepare statistical and management reports

3.The Proposed System

3.1 User requirement

Pongpej apartment is an apartment on Ngamwongwan road. It will open in march 1998. The main business is room rent, laundry and food service. ARS(apartment rental system) works on client server distributed system concept. Now the software cost is higher. Most of the total cost of computer in the organization is the software cost. The question is how to reduce the software cost in organization. The system is designed to use web server, database server , and web browser in the client machine with the internet communication with no need more application cost. Because we wish to reduce the transaction to as little as possible so we use the object oriented design project for reusing the data and separating all design by object oriented method for reusing the code too. ARS is to be created for an apartment. The rooms are devised as follows:

ROOM TYPE	ROOM NO SERIES / NAME	RATE / NIGHT
SINGLE	101 TO 130, 201 TO 230	350
TWIN	131 TO 145, 231 TO 252	500
SUITE	301 MAJESTIC SUITE 302 ROYAL SUITE 303 HONEYMOON SUITE	700

Figure 3.1 Room type and room rate

Reservation clerk receives request by phone or fax. He asks the customer what the required period of stay is i.e. the check in date and the check out date and type of room, number of rooms. He checks in his reservation to see the number of reservations that has already been made do not exceed the number and type of rooms required.

Counting the total number of rooms that is required by reservation satisfy the following conditions

1. The number of rooms required in the existing reservations having dates that fall within the required period; or
2. The required period is within the period of an existing reservation. This means the required period falls within the period of an existing reservation.

If the number of the above type of reservations is less than the total number of rooms of that type, he inform the customer that it is acceptable. He then asks if the reservation is personal or corporate. He also informs him of the different rates. Certain companies maintain corporate accounts with the apartment. These corporate customers are given certain discounts.

The following information is obtain from the customer. The customer may make reservations for a mixture of room types. If the customer is also a corporate customer, the clerk gets the company information. Then he registers the reservation. He also prepares the list of expected checked in reservations and inform the cleaning service to prepare as neccessary. The list of rooms expected to checked out the following day is also passed to the cleaning department. He also passes the information to the front desk and the accounts.

When the customer checks in, The front desk clerk inquires information about his reservation. I first check the list of expected check in passed to me by the reservation clerk. If the reservation is found, I will ask the customer for the type of payment he will be making. I will ask them to register themselves before giving the them room keys. Sometimes people may want to stay without making previous reservation. I will then carry out the reservation process to determine in which room he can stay.

-Payment types come in the following forms.

PAYMENT TYPE	POSSIBLE DISCOUNT
CASH	NO DISCOUNT
AGENCY'S CHECK	15% e.g. TRAVELING AGENCY
CREDIT CARD	10% e.g. AMEX, DINERS, VISA
CORPORATE	DEPENDENT UPON THE COMPANY

Table 3.2 Discount rate

After registering the customer and giving them the room keys, I will keep the list of room registration information and pass one set to the accountants, the cleaning and reservation clerks.

Customers may check out any time during their stay. Normally, the accountants pass me the bills for the expected checkouts. I may modify the bill or even create a new bill for last minute charges like telephone calls. I will check the room registration for the type of payment take into account the

discounts if any. Customers may also be charged service items to their bills e.g. Laundry or meals

taken at the hotel restaurant. These will also be part of the charge. I will then bill the customer for all of

these. After the clients have paid for their stay, the list of receipts is passed to the accountant. I also

inform the cleaning department of the list of check out rooms.

It is possible that the customer may want to extend his stay. I will then repeat the reservations process

changing the expected check out date if vacancy is available.

Account clerk After the reservation clerk passes the list of expected check in and check outs, he prepares an account for each expected check in. After the front desk informs me of the actual room assigned, I will update the account information with the room number.

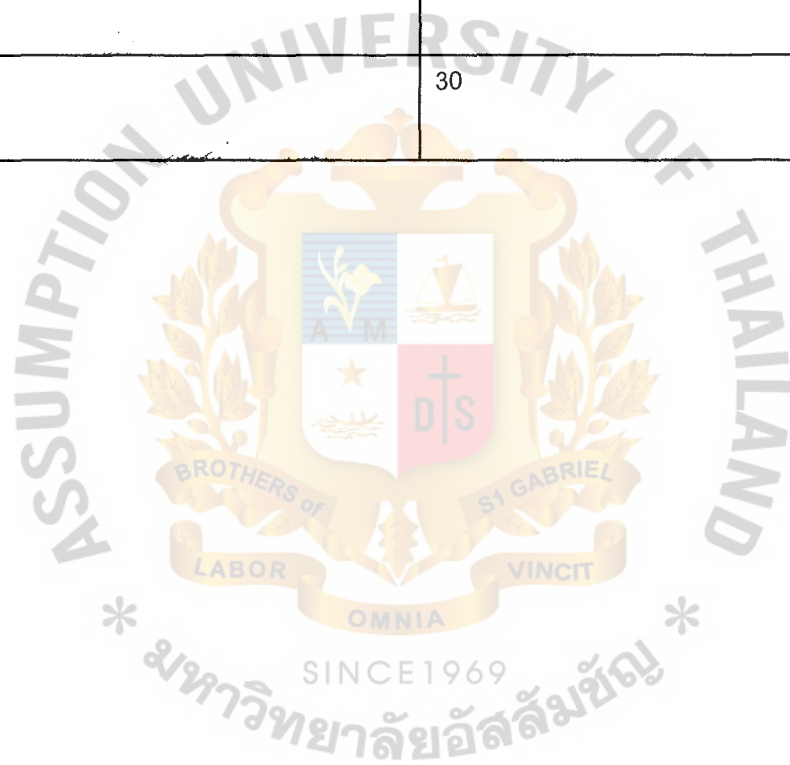
For the check outs, I prepare the room and service charges in the bill and pass it to front desk. When the food and beverage or cleaning section informs me of the services used, I update the guest's account accordingly.

Food and beverages supervisor guests may take meals in the restaurant or order meals to their rooms. These can be charged to the bill or paid by cash. If they are charged to the room, I will specify the detail and pass it to the accountants.

Cleaning supervisor Based on the expected check in and check out for the rooms, I assign the cleaning responsibility to the staff. The guest may also request for laundry service by sending the items to be cleaned. I will then charge for the service and inform the accountants. The following rates are for the laundry service:

Table 3.3 Laundry rate

CLOTH TYPE	RATE / PIECE
SHIRT	15
TROUSERS / BLOUSE	25
COATS	50
TOWER	30



3.2 Systems Design

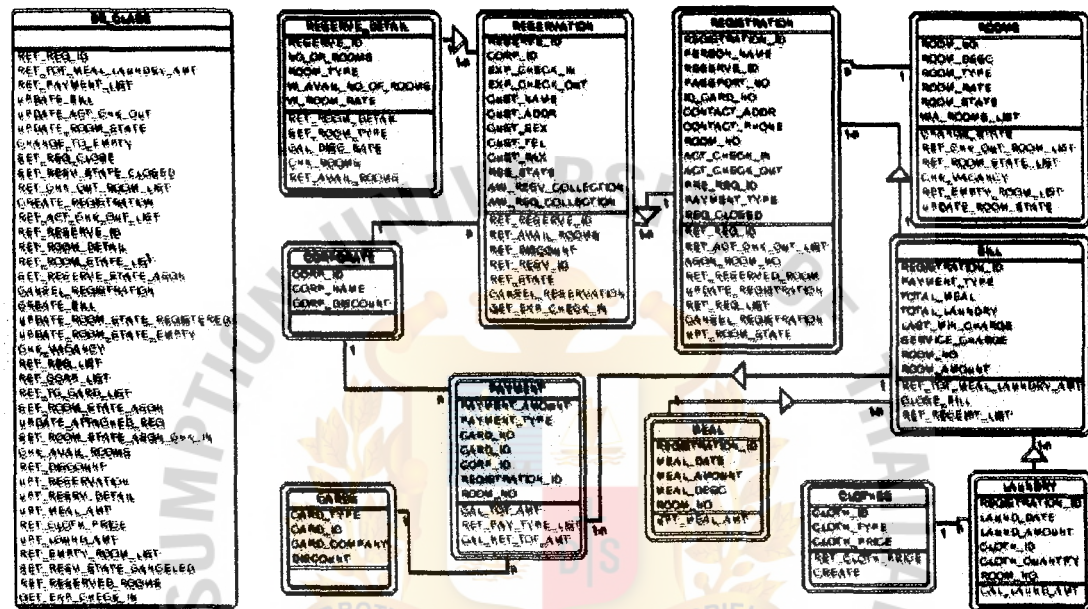


Figure 3.1 Apartment rental system objects model diagram

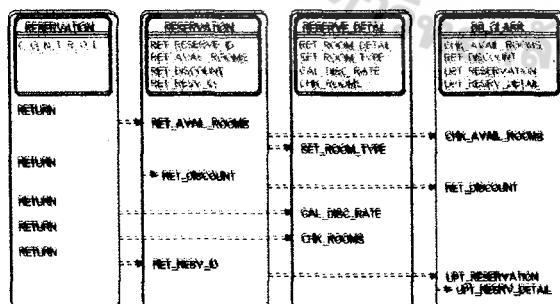
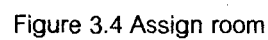
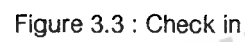


Figure 1.2 : RESERVATION



ARLO PREPARED ROOM NUMBER

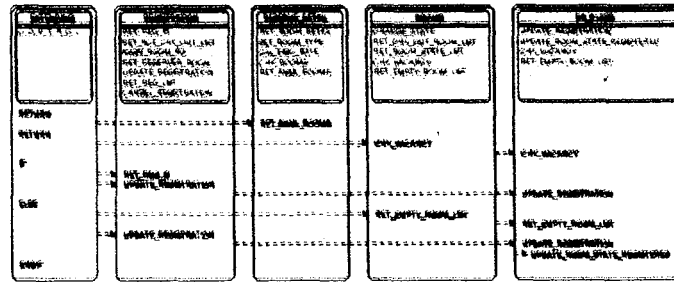


Figure 3.5 Extended room

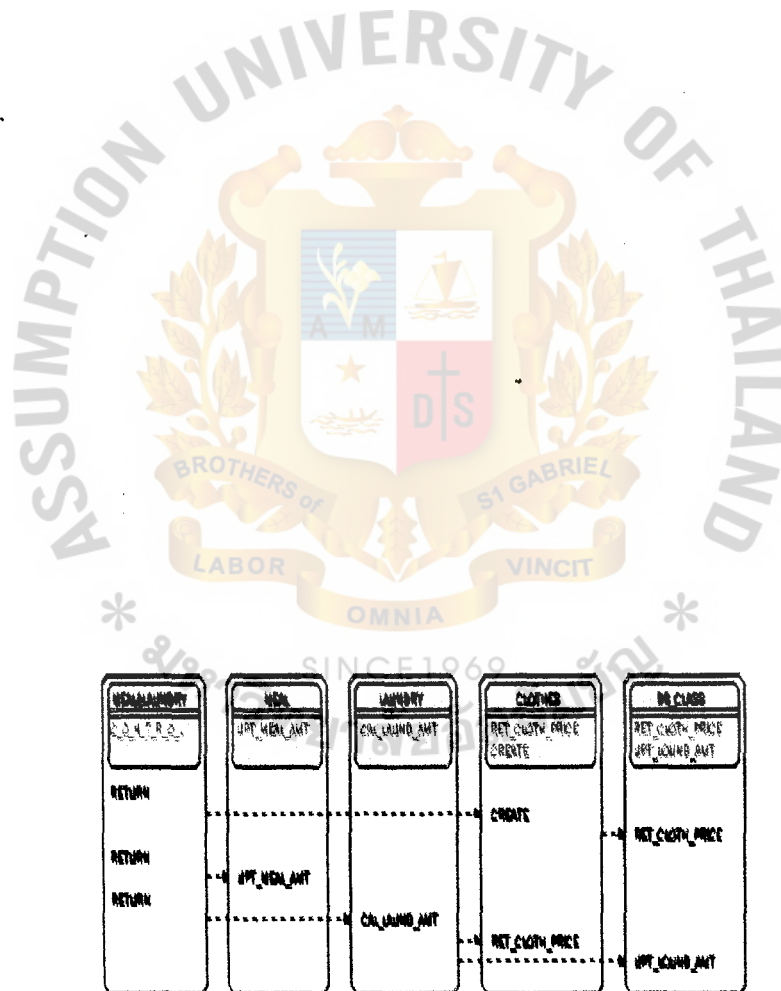
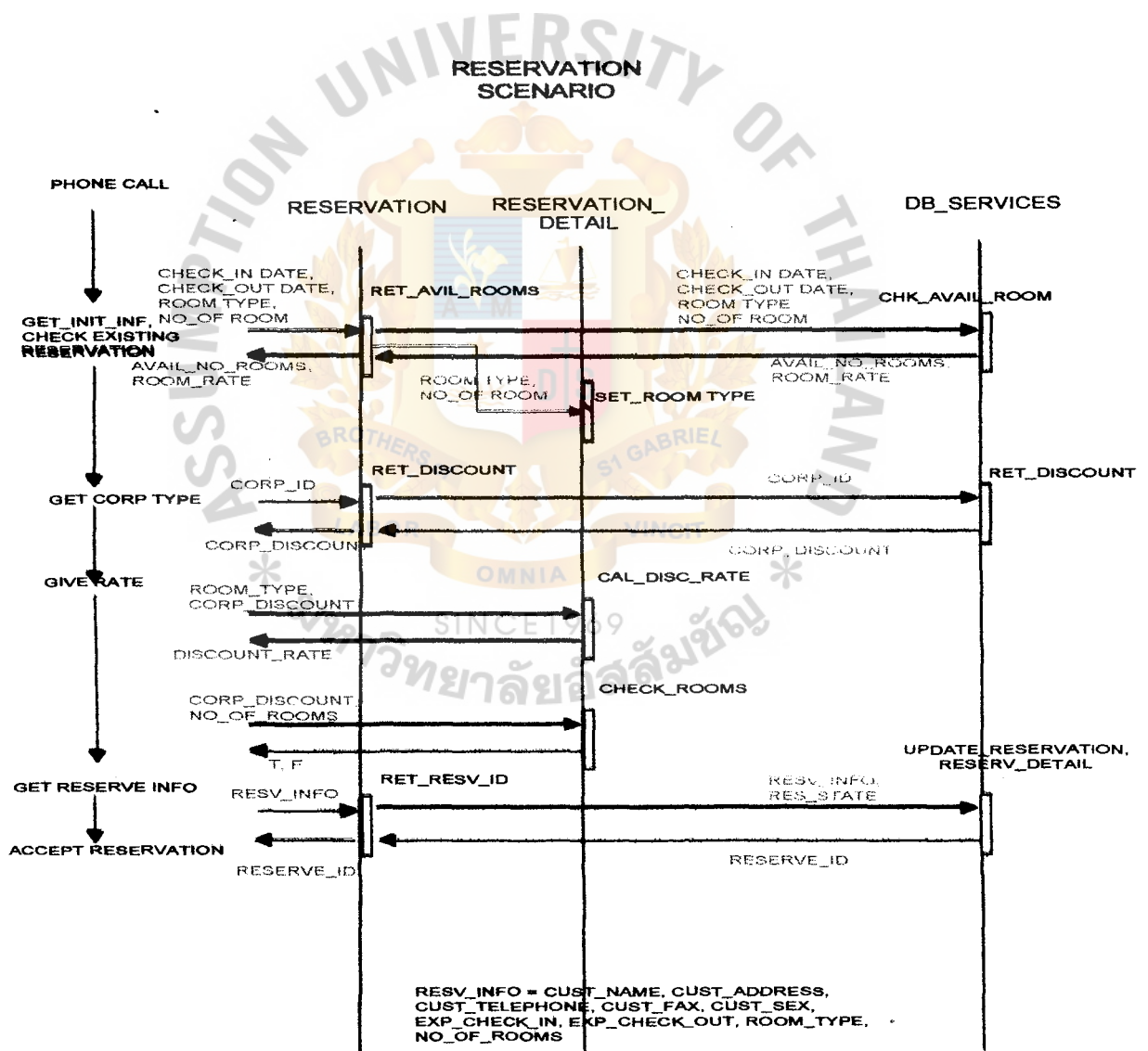


Figure 3.6 : Meal and laundry

3.2.1 Scenario Design

Figure 3.9 Reservation scenario



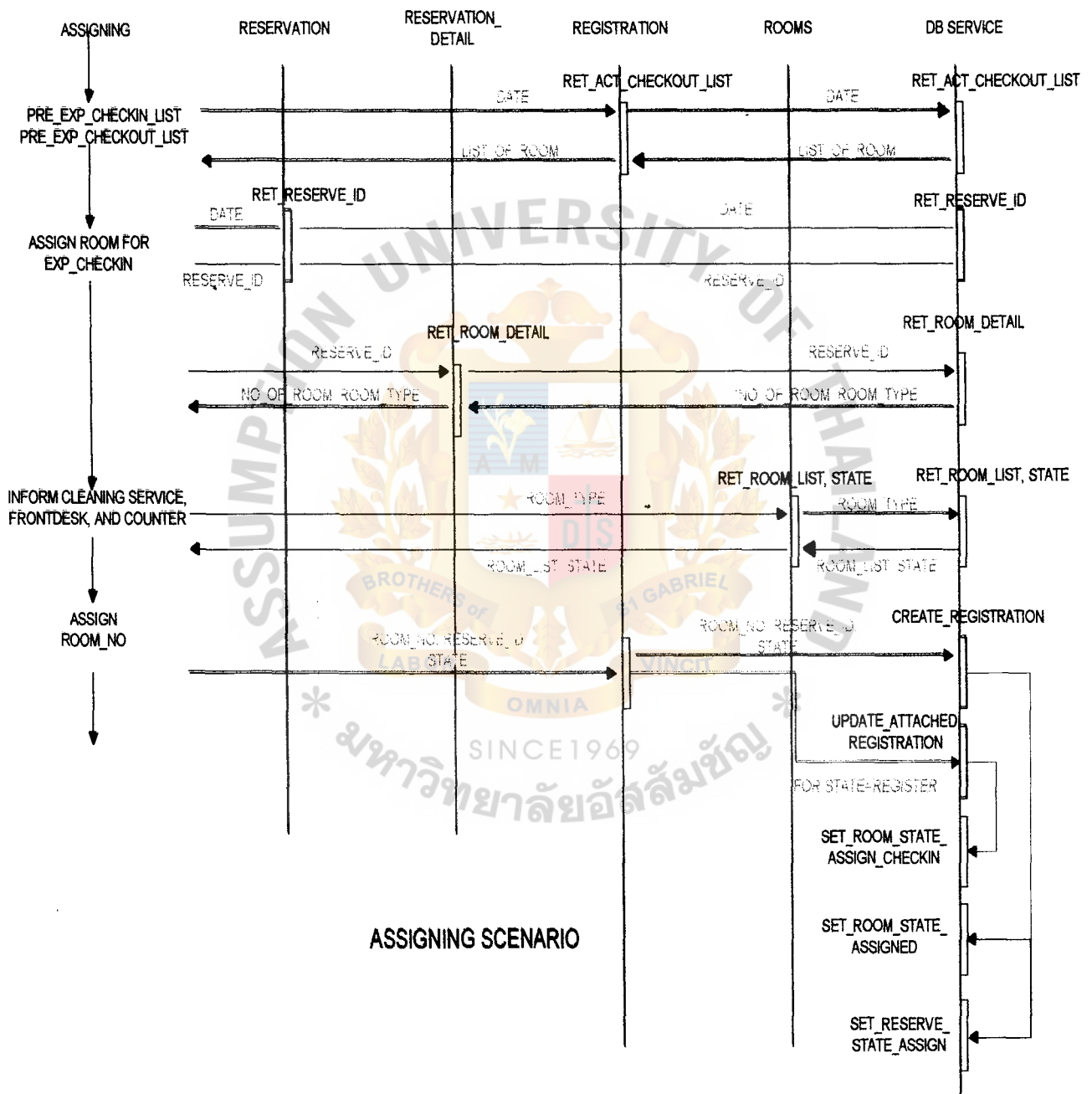


Figure 3.10 Assign scenario

CHECK IN SCENARIO

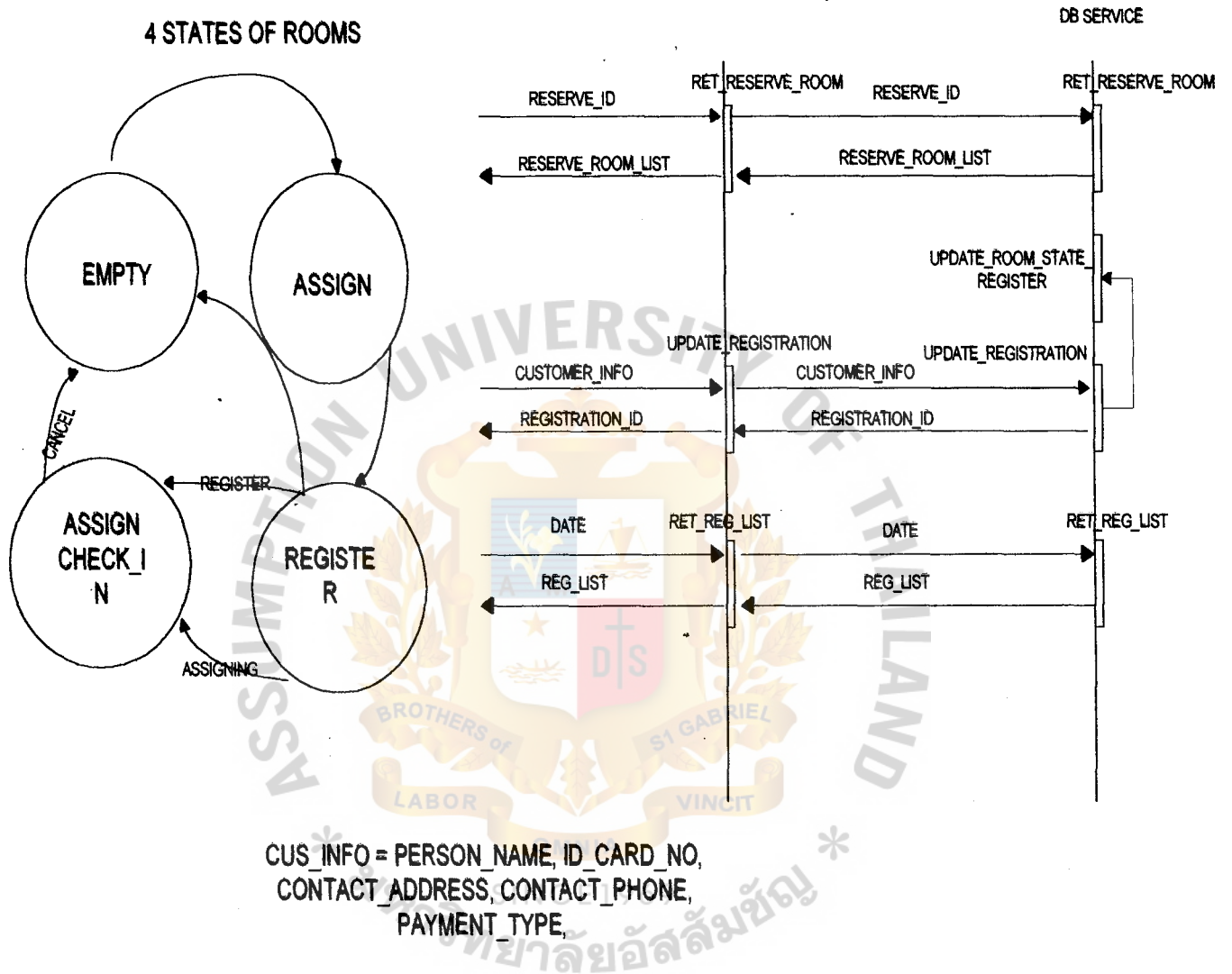


Figure 3.11 Check in scenario

EXTENDED ROOM SCENARIO

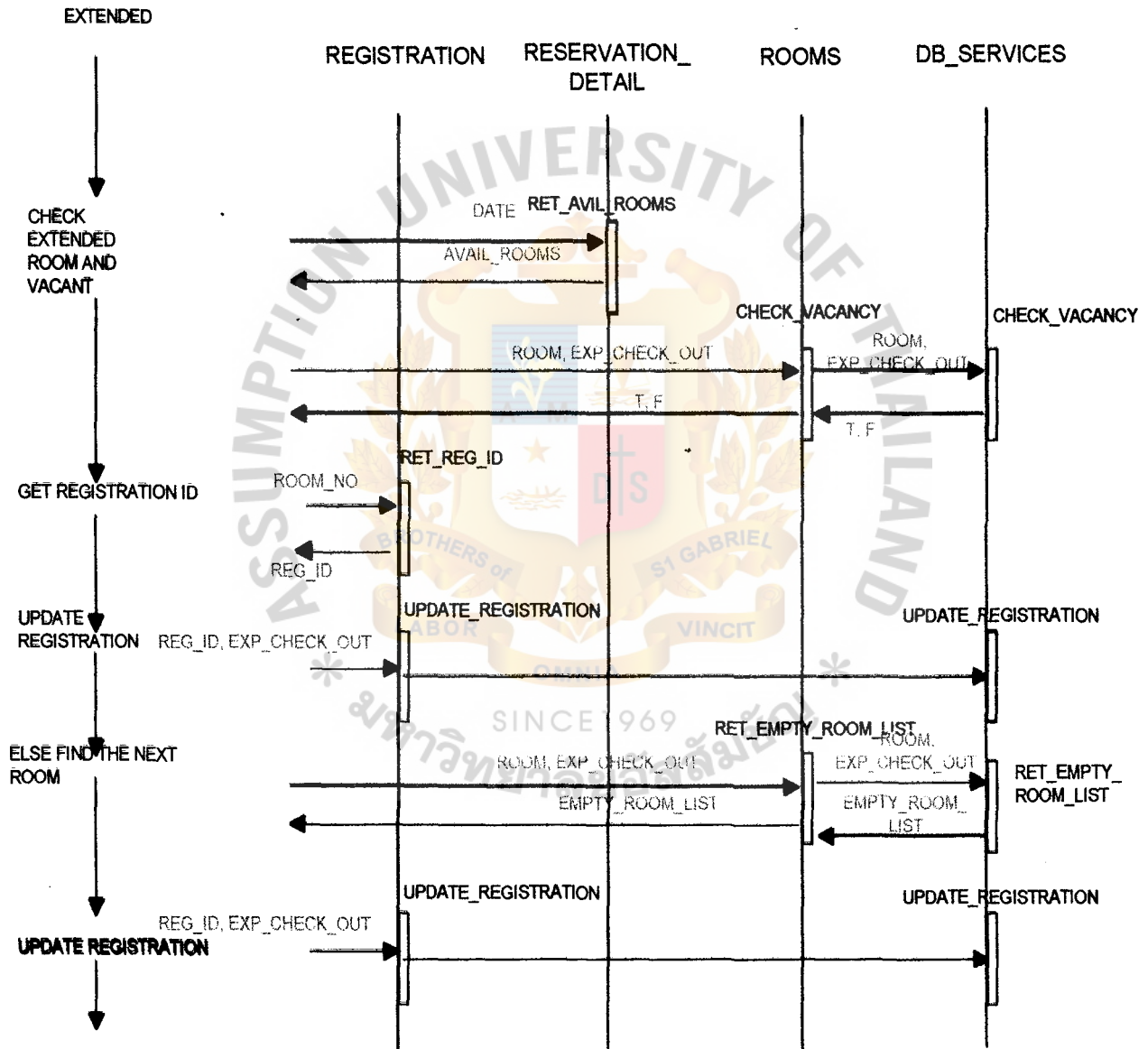


Figure 3.12 Extended scenario

CHECK OUT SCENARIO

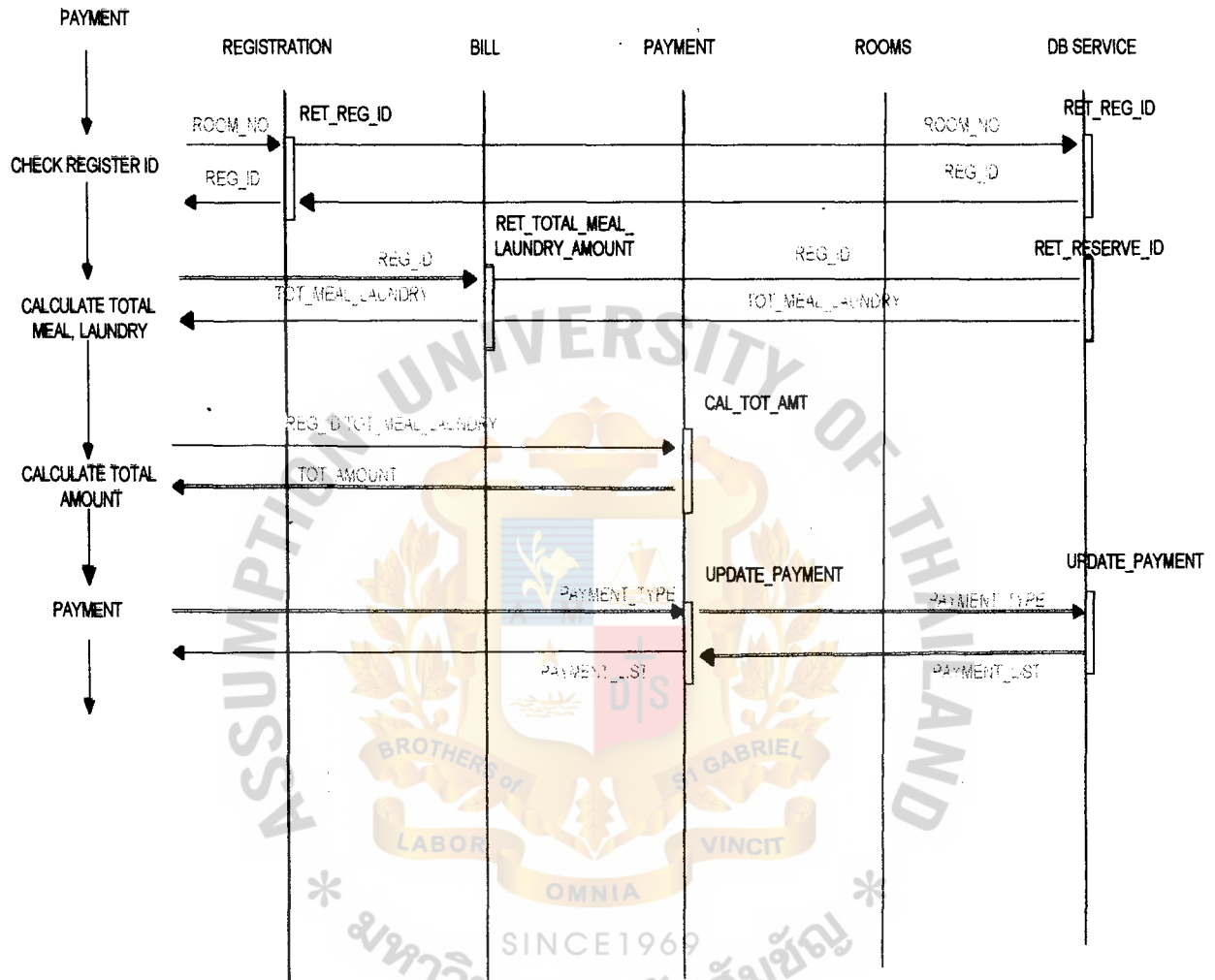


Figure 3.13 Check out scenario

3.2.2 Scenario Analysis

RESERVATION SCENARIO

RET_AVAIL_ROOMS(\$EXP_CHECK_IN, \$EXP_CHECK_OUT, \@ROOM_TYPE, \NO_OF_ROOMS,

\@WA_AVAIL_NO_OF_ROOMS, \@WA_ROOM_RATE, \@ACCEPTABLE)

RET_DISCOUNT(\$CORP_NAME, \ \$CORP_DISCOUNT)

CAL_DISC_RATE(\@ROOM_TYPE, \$CORP_DISCOUNT, \ \$DISCOUNT_RATE)

RET_RESV_ID(\%RESV_INF, \ \$RESERVE_ID)

RESV_INF{CUST_NAME}

RESV_INF{CUST_ADDRESS}

RESV_INF{CUST_TEL}

RESV_INF{CUST_FAX}

RESV_INF{CUST_SEX}

RESV_INF{CUST_CHECK_IN}

RESV_INF{CUST_CHECK_OUT}

END OF DAY BY RESERVATION CLERK

RET_ACT_CHECK_OUT_LIST(\$DATE, \@WA_ROOM_NO)

ASSIGNING

RET_RESERVE_ID_LIST(\$DATE, \@RESERVE_ID)

RET_ROOM_DETAIL(\@RESERVE_ID, \@NO_OF_ROOMS, \@ROOM_TYPE)

RET_ROOM_STATE_LIST(@ROOM_TYPE, \@ROOM_NO, \@ROOM_STATE)

ASSIGN_ROOM_NO(\@WA_ROOM_NO, \@WA_RESERVE_ID, \@WA_ROOM_STATE)

CHECK IN

RET_RESERVED_ROOM(\$RESERVE_ID, \@WA_ROOM_TYPE)

UPDATE_REGISTRATION(\%CUST_INFO, \$ROOM_NO, \$PAYMENT_TYPE, \$REGISTRATION_ID)

CUST_INFO{PERSON_NAME}

CUST_INFO{PASSPORT_NO}

CUST_INFO{ID_CARD_NO}

CUST_INFO{CONTACT_ADDRESS}

CUST_INFO{CONTACT_PHONE}

CUST_INFO{ACT_CHECK_IN}

CUST_INFO{ACT_CHECK_OUT}

END OF DAY BY FRONT DESK CLERK

RET_REG_LIST(\$DATE, @WA_ROOM_NO, @WA_REGISTRATION_ID)

CHECK OUT

RET_REG_ID(\$ROOM_NO, \$REGISTRATION_ID, \$ROOM_TYPE, \$ACT_CHECK_IN,

\$PAYMENT_TYPE,

\$ROOM_RATE, \$PERSON_NAME)

RET_TOT_MEAL_LAUNDRY_AMT(\$REGISTRATION_ID, \$TOTAL_MEAL, \$TOTAL_LAUNDRY)

CAL_TOT_AMT(%BILL_INFO, \$PAYMENT_AMOUNT, \$ROOM_CHARGE, \$SERVICE_CHARGE,

\$TAX)

BILL_INFO{TOTAL_MEAL}

BILL_INFO{TOTAL_LAUNDRY}

BILL_INFO{LAST_MIN_CHARGE}

BILL_INFO{ROOM_RATE}

BILL_INFO{ACT_CHECK_IN}

BILL_INFO{DATE}

IF NOT CASH:

RET_PAY_TYPE_LIST(\$PAYMENT_TYPE, \@WA_CARD_COMPANY, \@WA_DISCOUNT)

CAL_NET_TOT_AMT(\$CARD_NO, \$DISCOUNT, \$COMPANY_NAME, \\$TOTAL_AMOUNT)

CLOSE_BILL(\$REGISTRATION_ID)

CHANGE_STATE(\$ROOM_NO)

CLOSE_REGISTRATION(\$REGISTRATION_ID)

CLOSE_PAYMENT(\$REGISTRATION_ID)

END OF DAY

RET_CHECK_OUT_ROOM_LIST(\$DATE, \@WA_ROOMS)

MEAL AND LAUNDRY

UPT_MEAL_AMT(\$ROOM_NO, \$MEAL_DATE, \$MEAL_AMOUNT)

CAL_LAUN_AMT(\$ROOM_NO, \$LAUND_DATE, \@CLOTH_ID, \@CLOTH_QUANTITY)

EXTENDING

CHECK_VACANCY(\$ROOM_NO, \\$VACANCY)

RET_EMP_ROOM_LIST(\$ROOM_TYPE, \@ROOM_LIST)

CANCEL

CANCEL_RESERVATION(\$RESERVE_ID, \ \$CANCELLED)

UPDATE_ROOM_STATE(\$RESERVE_ID)

CANCEL_REGISTRATION(\$RESERVE_ID, \$TYPE_OF_ROOM, \$NO_OF_ROOMS)



3.2.3 Screen Design

Figure 3.14 Table of contents (Start page)

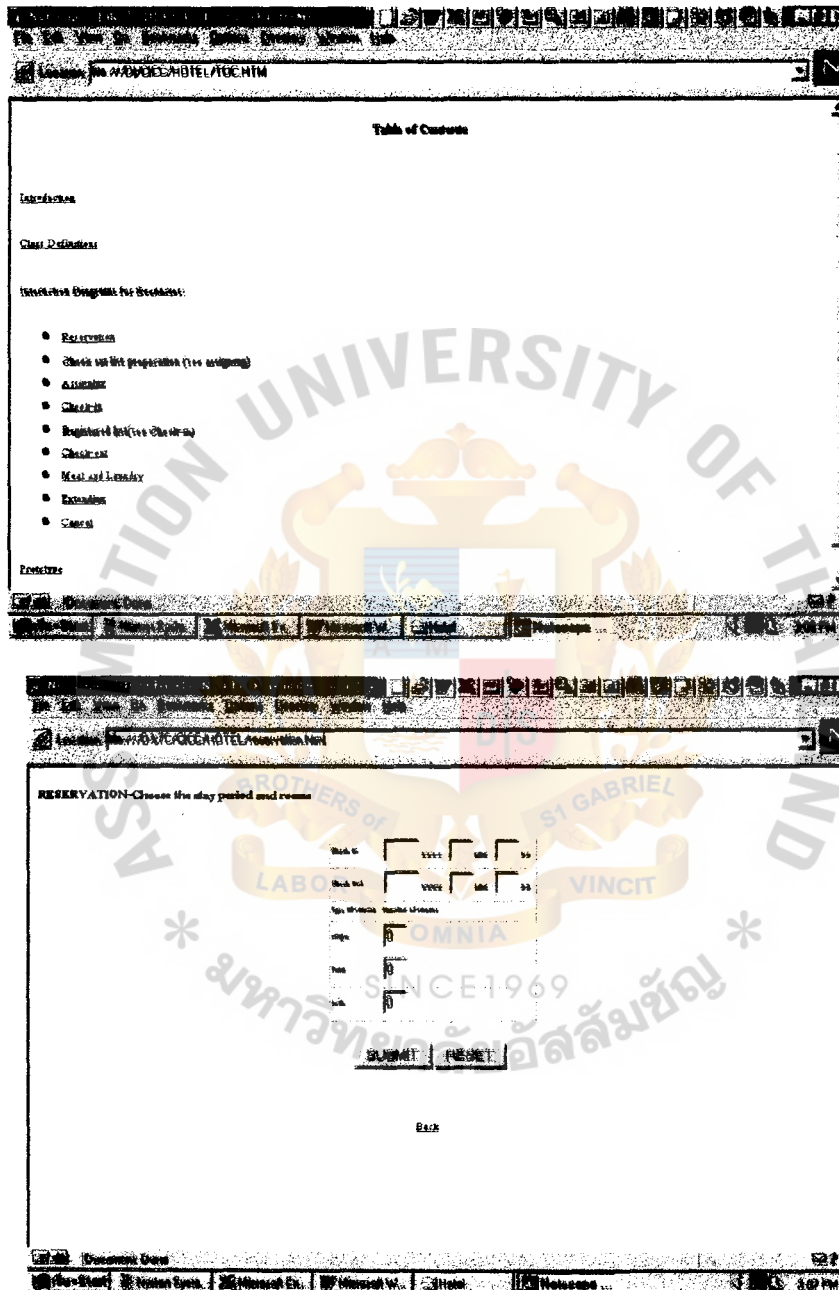


Figure 3.15 Reservation

EXPECTED CHECK OUT-Check out tomorrow

CHECK OUT DATE

[Submit](#) [Reset Form](#)

[Back](#)

[Copyright information goes here.](#)
Last revised: December 19, 1999

Figure 3.18 Cancel Reservation

ASSIGN-Get the reserve ID by date

ASSIGN DATE

[ASSIGN](#)

[Back](#)

[Copyright information goes here.](#)

Figure 3.19 Get Reservation by date

WATSONHOTEL.com

CHECK OUT REPORT - Check out report at the end of the day

DATE:

REGISTER NO	ROOM TYPE	ROOM NO.	CUSTOMER NAME	DEL CHECK IN	CHECK OUT	COMPANY

Back

WATSONHOTEL.com

Home System Data Microsoft Word - 2003 Notes Watsons - CHET... 2:29 PM

Figure 3.20 Check out report

WATSONHOTEL.com

CHECK OUT - Check out detail

REGISTRATION		ENTER IN DATE	
ROOM NO.		ENTER OUT DATE	
ROOM CODE		ENTER IN RATE	
NAME			

ROOM STATUS	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	
IN ROOM	

WATSONHOTEL.com

Home System Data Microsoft Word - 2003 Notes Watsons - CHET... 2:29 PM

Figure 3.21 Check out detail

File Edit View Options Database Window Help

Location: H:\WD\PC\OCC\HOTEL\rentout_get_room_no.htm

CHECK OUT - Check out room number

Room No. 101

LAST NEW CHANGE 102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

Submit Post Room

Back

File Edit View Options Database Window Help

Message: [CHECK...]

Figure 3.22 Check out gets room number

File Edit View Options Database Window Help

Location: H:\WD\PC\OCC\HOTEL\rentout_get_room_no.htm

FRONT DESK CLERK-Registration report

REGISTRATION DATE:

ROOM NO. REGISTERED:

Back to Menu

File Edit View Options Database Window Help

Message: [FRON...]

Figure 3.23 Registration report

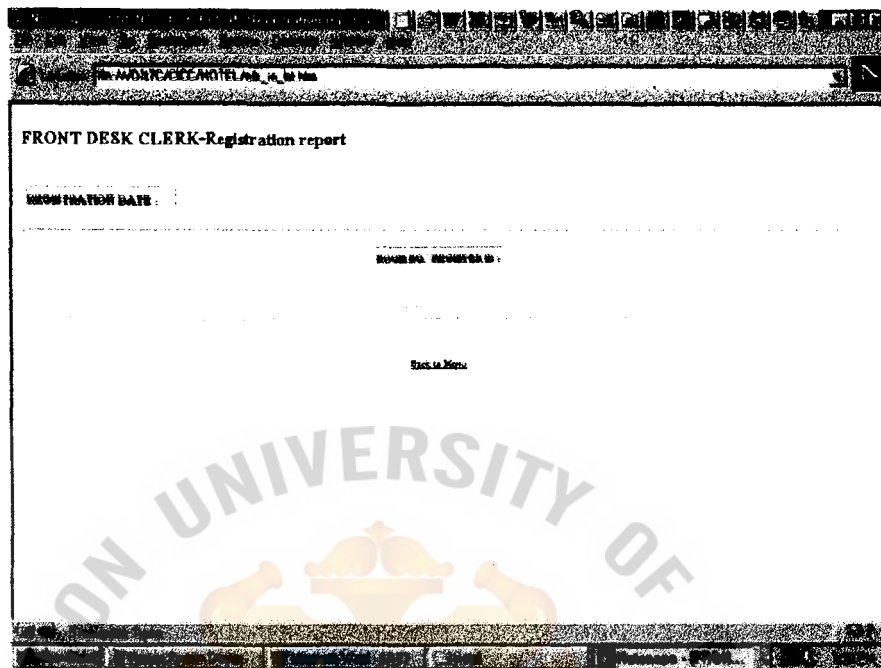


Figure 3.23 Registration report

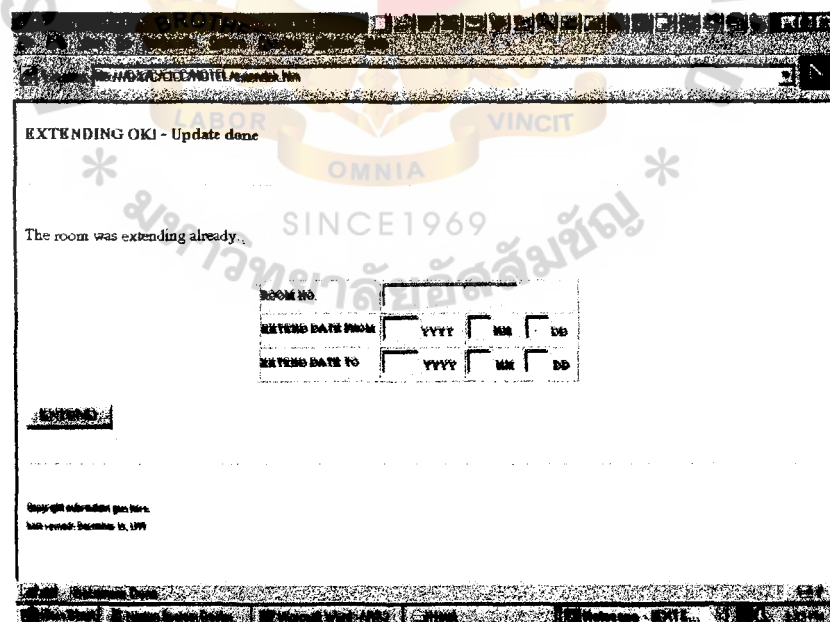


Figure 3.24 Extended room Update OK

FRONT DESK CLERK-Registration report

REGISTRATION DATE :

ROOM NO. NUMBER IN :

Back to Menu

Figure 3.23 Registration report

EXTENDING OK! - Update done

The room was extending already.

ROOM NO.	
EXTENDING DATE FROM	YYYY MM DD
EXTENDING DATE TO	YYYY MM DD

Copyright information here.
Last revised: December 15, 1991

Figure 3.24 Extended room Update OK

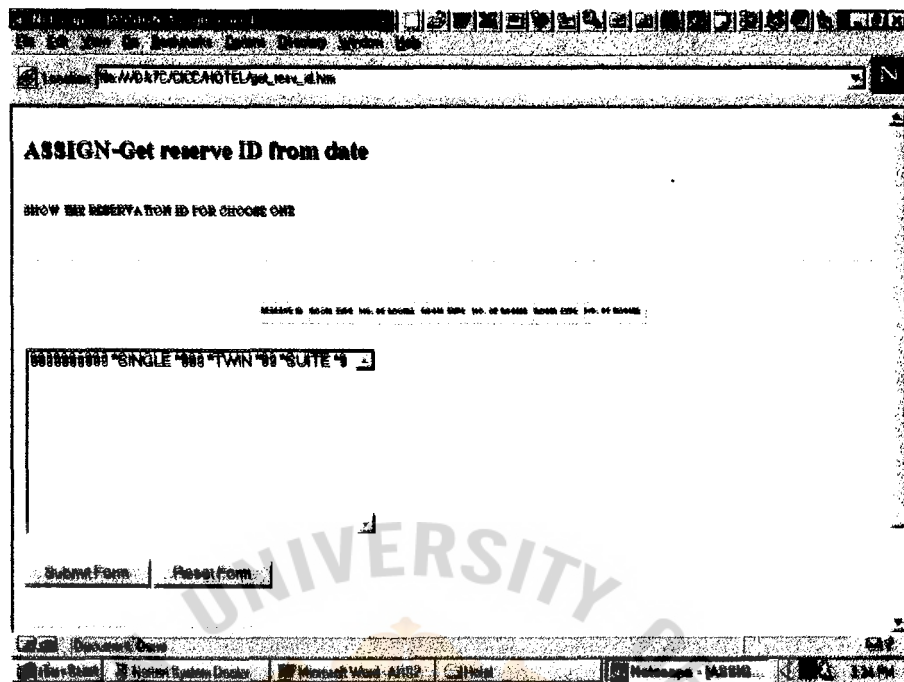


Figure 3.25 Report gets reserve ID form date



LAUNDRY-Lundry charge:

Room no. date

LAUNDRY CHARGE

LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00

[System](#) [Reset Form](#)

Back

Copyright information page here.
Last visited: December 13, 1999

Microsoft Word

Microsoft Word: APM... Netscape - LAUN... 3:26 PM

Figure 3.26 Laundry

LAUNDRY-Update OK!

Room no. date

LAUNDRY CHARGE

LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00
LAUNDRY	00

[Back to Menu](#)

Copyright information page here.
Last visited: December 13, 1999

Microsoft Word

Microsoft Word: APM... Netscape - LAUN... 3:26 PM

Figure 3.27 Laundry Update OK

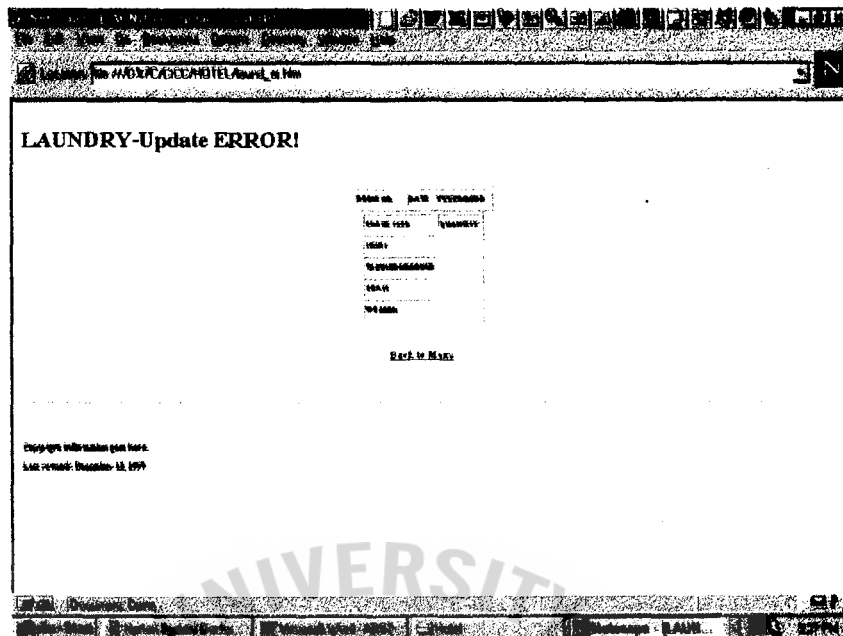


Figure 3.28 Laundry Update error

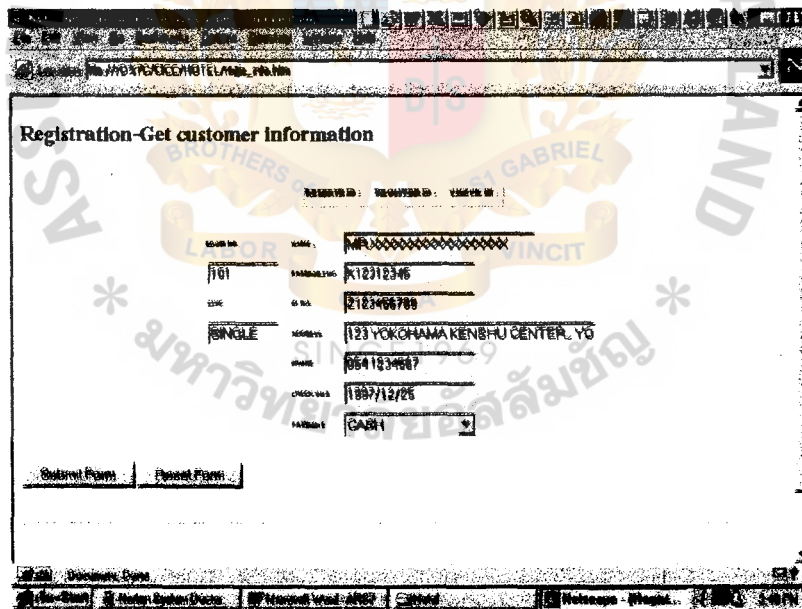


Figure 3.29 Registration gets customer information

Registration-Update OK!

RECEIVED IN	REGISTERED IN	VOICE IN
DATE IN	NAME	MR. XXXXXXXXXXXXXXXX
101	CLASSROOM NO.	K12312345
	AGE	2123456789
SINGLE	ADDRESS	123 YOKOHAMA KENSHU CENTER, YO
	PHONE	064 1234567
	EMAIL ADDR	1989/12/24
	AD METHOD	CASH

Back to Menu

Figure 3.30 Register update OK

Register-Ask reserve ID

Ask reserve ID for student information

Reserve ID: _____

Student: _____

Pass: _____

Back

Figure 3.31 Register Ask reserve ID

7. Dot Matrix Printer 2 Units

8. Tape Backup Drive 1 Unit

- Capacity 2 GB

9. UPS 3 Units

- Capacity 800VA

Software specification

1. Microsoft Window NT Server version 4.0 5 Users License

2. Microsoft Window 95 Thai Edition 4 Users License

3. Microsoft FrontPage 97

4. Microsoft Access 97

5. Perl Script Language

3.4 Security and controls

One of the most important considerations in system development and on going operation is the system security. Security of the system is concerned with the security of hardware and security of data.

Hardware security

Hardware security refers to all the equipment's related to the computer system such as the computer itself, harddisk, LAN card, network equipment, and other things that could be illegally accessed, stolen,

2. PC Client 4 Units

- Pentium 200MHZ Processor
- 16MB RAM
- Harddisk 1 GB
- Floppy Disk 1.44 MB
- VGA Controller
- 14" Super VGA Color Monitor
- Mouse and Keyboard

3. HUB 8 port

- 100 Mbps
- UTP interface port

4. LAN card 5 Units

- 10/100 Mbps
- UTP interface port

5. Unshield Twisted Pair (UTP) Cable

- category 5 (supporting 100Mbps)

6. Laser Printer

- 300-600 dpi

7. Dot Matrix Printer 2 Units

8. Tape Backup Drive 1 Unit

- Capacity 2 GB

9. UPS 3 Units

- Capacity 800VA

Software specification

1. Microsoft Window NT Server version 4.0 5 Users License
2. Microsoft Window 95 Thai Edition 4 Users License
3. Microsoft FrontPage 97
4. Microsoft Access 97
5. Perl Script Language

3.4 Security and controls

One of the most important considerations in system development and on going operation is the system

security. Security of the system is concerned with the security of hardware and security of data.

3.5 Cost / Benefit analysis

Cost analysis

The cost association with a new system consists of cost of building the system, cost of installing the system, cost of operation and maintaining the system.

Cost of building the system

- Cost of system analysis and requirement determination (2 weeks)	50,000
- Cost of system design (2 weeks)	50,000
- Cost of development and system testing (4 weeks)	40,000
Total cost of building the system	140,000

Cost of installing the system

- Cost of hardware

PC Server	1	Unit	300,000
PC Workstation	4	Units	200,000
Laser Printer (HP Laser Jet 5P)	1	Unit	26,000
Dot matrix printer (NEC P7300)	2	Units	40,000

UPS 800 VA	3	Units	18,000
LAN card	5	Units	15,000
HUB 8 ports	1	Unit	25,000
Tape Backup drive	1	Unit	13,000
- Cost of site preparation (Network installation)			
- Cost of software			
Microsoft Window NT Server V.4	4	Users license	23,000
Microsoft Window 95 Thai Edition	4	Users license	0
Microsoft FrontPage97			17,000
Microsoft Access 97			25,000
- Cost of implement and training (1 week)			25,000
Total cost of installing the system			724,500
- Cost of operation and maintaining the system (first year)			
- Computer accessories and supplies			24,000
- Program maintenance			14,000
Total cost of operation and maintaining the system			38,000
cost of the system			902,500

- Cost of site preparation (Network installation)

- Cost of software

Microsoft Window NT Server V.4	4	Users license	23,000
--------------------------------	---	---------------	--------

Microsoft Window 95 Thai Edition	4	Users license	0
----------------------------------	---	---------------	---

Microsoft FrontPage97			17,000
-----------------------	--	--	--------

Microsoft Access 97			25,000
---------------------	--	--	--------

- Cost of implement and training (1 week) 25,000

Total cost of installing the system	724,500
-------------------------------------	---------

- Cost of operation and maintaining the system (first year)

- Computer accessories and supplies 24,000

- Program maintenance 14,000

Total cost of operation and maintaining the system	38,000
--	--------

Total cost of the system	902,500
--------------------------	---------

Benefit analysis

Year	Systems Cost	System Benefits	Net Cumulative
1	902,500	360,000	-542,5000
2	38,000	360,000	-220,000
3	38,000	360,000	101,500
4	38,000	360,000	423,500
5	38,000	360,000	745,5000
Total	1,054,500	1,800,000	

Table 4.1 Compare cost / benefit for 5 years

Year	Systems Cost	System Benefits	Net Cumulative
1	902,500	360,000	-542,5000
2	38,000	360,000	-220,000
3	38,000	360,000	101,500
4	38,000	360,000	423,500
5	38,000	360,000	745,5000
Total	1,054,500	1,800,000	

Table 4.1 Compare cost / benefit for 5 years

4. Project Implementation

4.1 Implementation procedures

Implementation includes all the activities that are related to preparing the system for end users. The

following activities are carried out in the implementation phase:

- Site preparation
- Hardware installation
- Software installation
- User training
- Data conversion
- Parallel run
- System review
- Live run
- Post implementation review

Site preparation

Site preparation involves the installation of electrical and networks equipment. Electrical installation

includes the electrical wiring and placement of outlets, if they are not already in place. Then the UPS is

installed and tested. Network installation includes the network wiring and placement of outlets. These activities take 3 days to complete.

Hardware installation

After the site preparation is completed, all the hardware will be put in place and tested. This can be done in a few hours.

Software installation

The installation of the software includes installing operation system, Windows NT for server and Windows 95 for clients, Microsoft Access database, FrontPage and perl library on the server. The network system is tested to ensure that the client workstations can access programs and database on the server. Then users and security scheme are created and set up in the system. This takes 3 days.

User training

User training can be done prior to or in parallel with the three activities above. The persons that will be associated with or affected by the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. Both systems operators and users need training.

-**System operators** are persons who are responsible for keeping the equipment's running as well as for providing the necessary support service. Their training must ensure that they are able to handle all possible operations. Those persons are such as EDP manager or computer operator. The training needed for system operators are Windows NT, Windows 95, Internet Explorer and Apartment Rental System.

- **Users** are persons who will be using the system in assistance to their work. User training involves equipment usage and how to operate the system implemented. The operation of system training emphasizes on the data handling activities and procedures. This includes adding data or new transaction, editing data, formulating questions, deleting data and producing report in all the modules in the system.

Data conversion or input initial data into the system

If there is an existing system in operation, all the data fields in files or tables must be identified. Then they must be compared with the data structure of the new system so that the correct data are transferred into the new system. For the data fields that do not exist in the existing system and required by the new system, it is the responsibility of users to define the values of those data and input them into the system. If data is not too much. The data can be manually input to the new system. This method is highly recommended as to get users familiar with the system and validation of data is properly done

through the entry procedure of new system. But in the case where there are too many data to handle, a conversion program will be developed to get the data from the existing system and transfer to the new system.

Test run (parallel run)

When all the data are entered into the system and everything is in place, the system is ready to run. At the beginning, the system should be operated in parallel with the existing system. Since users are not yet familiar with the new system and unexpected problem could occur. This step should go on for around 2 weeks or until users are confident with the new system.

System review

System review is carried out throughout the test run of the new system. The system review is the process conducted by user and system analysis to determine how well the system is working, how it has been accepted and whether adjustments are needed. If any unexpected error occurs, they can be corrected before the system goes live.

Live run

Finally, after the revision of the system and all the errors have been corrected, the system is now ready

to be turned over to the users and go on live.

Post Implementation Review

Post implementation review is a critical examination of the system after it has been put into production.

The evaluation is conducted three months after the live run of the system. The post implementation

reviews focus on the following:

- A comparison of the system's actual performance and the anticipated performance objectives.
- Errors and unexpected problems that occurred during the production
- Actual cost of running the system in comparison of the anticipated cost.

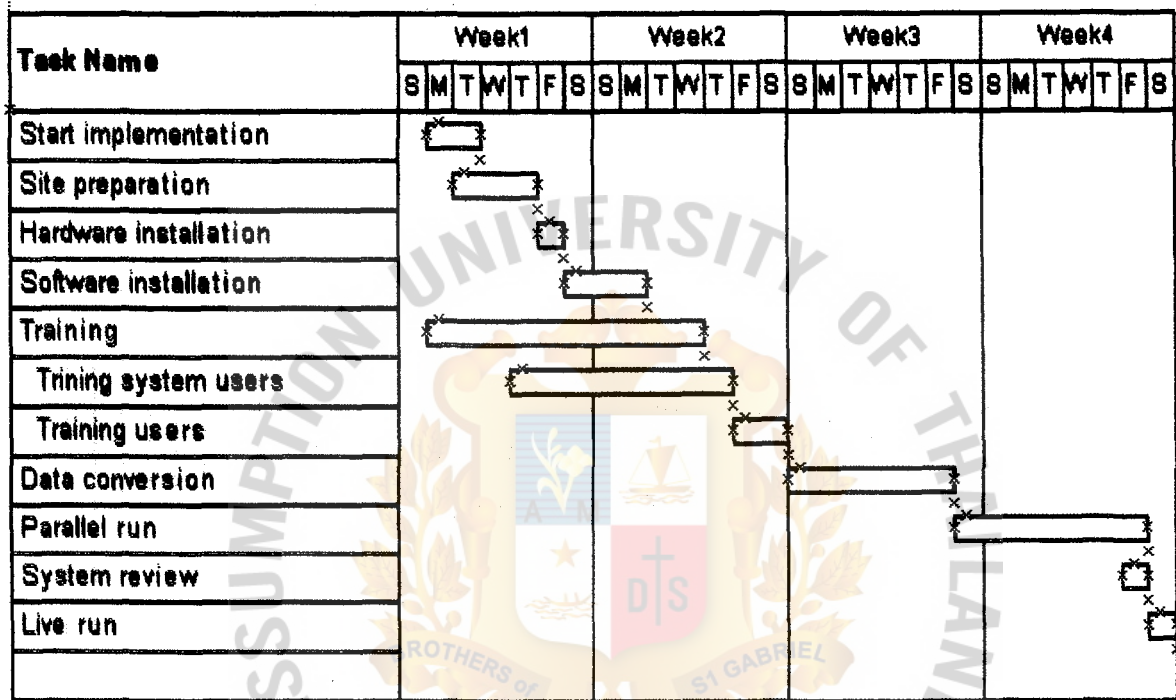


Figure 4.1 Project implement schedule

5. Conclusions and Recommendation

The study of system development project of Apartment Rental System reveals a need for a hotel organization to utilize a computer system. A system which automates the front desk and reservation processes, offers users a great deal of benefit, comparing to the manual system. A clear tangible benefit that an apartment organization receives from using the computer system is reducing the cost of manpower. The computerized system also provides a lot of intangible benefits, in terms of data accuracy, data security, efficiency and control, to the organization. Moreover, tasks, which are tedious, complex and repetitive, can be eliminated.

The system presented in this project was designed for an apartment organization in general. Some modification may be required in order for the system to fit user requirements of a specific organization.

Well-known and popular tools are used to develop the system. This ensures that the system tools will not be obsolete in near the future, and can be upgraded to a newer version. The operating system and network operation system are also easy to use and maintain. The hardware suggested computers and network equipment, are at the high end of its range. This is because the hardware technology changes very fast and the system will be able to cope with future expansion;

Although, the system was designed to assist users in almost all the functions and processes, still enhancements can be added to the system to provide even greater benefits to users. The following features are the recommendation of enhancements that should be added to the system.

- Interface and link to the telephone system for automatic calculation of expenses.
- Graphical layout of apartment's in hotel for convenience in room assignment task.
- A customer's history module for future room sales and analysis purpose.
- Analysis report for canteen department; and
- A module that fully handles organization of seminar or conference rooms.



References

1. Foster, Dennis L., Front Office Operations and Administration, Singapore : McGraw-Hill Book Co., 1991
2. Gray, William S., Hotel and Motel Management and Operation, NJ : Prentice-Hall, 1994
3. Kasavana, Michael L., Hotel Information Systems, N.Y. : Van Nostrand Reinhold, 1978
4. Kendall Kenneth E. and Kendall Julie E., System Analysis and Design, NJ : Prentice-Hall, 1992
5. Lawrence, Michael J., Information System Design, NJ : Prentice-Hall, 1982
6. Long, Larry., Management Information Systems, NJ : Prentice-Hall, 1989
7. Page-Jones, Meilir., The Practical Guide To Structured System Design, NJ : Prentice-Hall, 1988
8. Rutherford, Denney G., Hotel Management and Operations, N.Y. : Van Nostrand Reinhold, 1995
9. Seen, James A., Analysis and Design of Information Systems, Singapore : McGraw-Hill Book Co., 1989
10. Yourdon, Edward., Modern Structured Analysis, Englewood Cliffs, NJ : Prentice-Hall, 1989



Appendix A

Class and Function Design

CLASS NAME	BILL	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS CONTAINS BILLING INFORMATION		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	REGISTRATION_ID	REGISTRATION IDENTIFICATION
PRIVATE	STRING	PAYMENT_TYPE	THE MODE OF PAYMENT
PRIVATE	STRING	TOTAL_MEAL	TOTAL MEAL USE BY CLIENT
PRIVATE	STRING	TOTAL_LAUNDRY	TOTAL LAUNDRY USE BY CLIENT
PRIVATE	INTEGER	LAST_MIN_CHARGE	CHARGES FOR LAST MINUTE SERVICES
PRIVATE	INTEGER	SERVICE_CHARGE	CHARGES FOR SERVICES
PRIVATE	STRING	ROOM_NO	ROOM NUMBER
PRIVATE	INTEGER	ROOM_AMOUNT	ROOM AMOUNT
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	RET_TOT_MEAL_LAUNDRY_AMT		
PUBLIC	CLOSE_BILL		
PUBLIC	RET_RECEIPT_LIST		

Table A-1 CLASS BILL

CLASS NAME	CARDS	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS DESCRIBE INFORMATION ABOUT THE CREDIT CARDS		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	CARD_TYPE	TYPE OF CREDIT CARD
PRIVATE	STRING	CARD_ID	IDENTIFICATION OF CREDIT CARD
PRIVATE	STRING	CARD_COMPANY	CREDIT CARD COMPANY
PRIVATE	INTEGER	DISCOUNT	DISCOUNT OF CREDIT CARD
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	

Table A-2 CLASS CARDS

CLASS NAME	CLOTHES	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS CONTAINS CLOTHES INFORMATION		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	CLOTH_ID	CLOTH IDENTIFICATION
PRIVATE	STRING	CLOTH_TYPE	TYPE OF CLOTH
PRIVATE	INTEGER	CLOTH_PRICE	PRICE FOR CLEANING DIFFERENT CLOTHES
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	RET_CLOTH_PRICE		
PUBLIC	CREAT		

Table A-3 CLASS CLOTHES

CLASS NAME	CORPORATE	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS REFERENCE TABLE THAT KEEPS CORPORATE INFORMATION		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	CORP_ID	CORPORATE IDENTIFICATION
PRIVATE	STRING	CORP_NAME	NAME OF CORPORATE
PRIVATE	INTEGER	CORP_DISCOUNT	CORPORATE DISCOUNT
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	

Table A-4 CLASS CORPORATE

CLASS NAME	LAUNDRY	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS DESCRIBE INFORMATION ABOUT LAUNDRY SERVICES		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	REGISTRATION_ID	REGISTRATION IDENTIFICATION
PRIVATE	STRING	LAUN_DATE	DATE OF ORDER FOR LAUNDRY
PRIVATE	INTEGER	LAUNDRY_AMOUNT	AMOUNT FOR LAUNDRY SERVICES
PRIVATE	STRING	CLOTH_ID	CLOTH IDENTIFICATION
PRIVATE	STRING	CLOTH_QUANTITY	QUANTITY OF CLOTH CLEANED
PRIVATE	STRING	ROOM_NO	ROOM NUMBER
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	CAL_LAUND_AMT		

Table A-5 CLASS LAUNDRY

CLASS NAME	MEAL	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS CONTAINS MEAL INFORMATION		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	REGISTRATION_ID	REGISTRATION IDENTIFICATION
PRIVATE	DATE	MEAL_DATE	DATE OF ORDER FORMEAL
PRIVATE	INTEGER	MEAL_AMOUNT	AMOUNT FOR MEAL
PRIVATE	STRING	MEAL_DESC	DESCRIPTION FOR MEAL
PRIVATE	STRING	ROOM_NO	ROOM NUMBER
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	UPT_MEAL_AMT		

Table A-6 CLASS MEAL

CLASS NAME	PAYMENT	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS CONTAINS PAYMENT INFORMATION		
DATA/MEMBERS INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	INTEGER	PAYMENT_AMOUNT	TOTAL AMOUNT PAYABLE
PRIVATE	STRING	PAYMENT_TYPE	MODE OF PAYMENT
PRIVATE	STRING	CARD_ID	CREDIT CARD IDENTIFICATION
PRIVATE	STRING	CARD_NO	CREDIT CARD NUMBER
PRIVATE	STRING	CORP_ID	CORPORATE IDENTIFICATION
PRIVATE	STRING	REGISTRATION_ID	REGISTRATION IDENTIFICATION
PRIVATE	DATE	MEAL_DATE	DATE OF ORDER FORMEAL
PRIVATE	STRING	ROOM_NO	ROOM NUMBER
PRIVATE	STRING	PAY_CLOSE	PAYMENT CLOSED
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	CAL_TOT_AMT		
PUBLIC	RET_PAY_TYPE_LIST		
PUBLIC	CAL_NET_TOT_AMT		
PUBLIC	CLOSE_PAYMENT		

Table A-7 CLASS PAYMENT

CLASS NAME	REGISTRATION	DATE	1/5/1998
VERSION	1		
BASE CLASS	RESERVATION		
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS IS A TABLE KEEPS REGISTRATION INFORMATION		
DATAMEMBERS/ INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	REGISTRATION_ID	ID OF REGISTRATION
PRIVATE	STRING	PERSON_NAME	PERSON NAME
PRIVATE	STRING	RESERVE_ID	RESERVATION ID
PRIVATE	STRING	PASSPORT_NO	PASSPORT NUMBER
PRIVATE	STRING	ID_CARD_NO	IDENTIFICATION CARD IDENTIFICATION
PRIVATE	STRING	CONTACT_ADDR	CONTACT ADDRESS
PRIVATE	STRING	CONTACT_PHONE	CONTACT PHONE
PRIVATE	INTEGER	ROOM_NO	ROOM NUMBER
PRIVATE	DATE	ACT_CHECK_IN	ACTUAL CHECK IN DATE
PRIVATE	DATE	ACT_CHECK_OUT	ACTUAL CHECK OUT DATE
PRIVATE	STRING	PRE_REG_ID	PRE_REG_ID
PRIVATE	STRING	PAYMENT_TYPE	PAYMENT TYPE
PRIVATE	STRING	REG_CLOSED	REGISTRATION CLOSED
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	RET_REG_ID		
PUBLIC	RET_ACT_CHK_OUT_LIST		
PUBLIC	ASGN_ROOM_NO		
PUBLIC	RET_RESERVED_ROOM		
PUBLIC	UPDATE_REGISTRATION		
PUBLIC	RET_REG_LIST		
PUBLIC	CLOSE_REGISTRATION		
PUBLIC	UPT_ROOM_STATE		
PUBLIC	CANCEL_REGISTRATION		

Table A-10 CLASS REGISTRATION

CLASS NAME	RESERVE_DETAIL	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS IS A TABLE THAT KEEPS RESERVATION DETAIL		
DATAMEMBERS/ INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	RESERVE_ID	RESERVATION ID
PRIVATE	STRING	NO_OF_ROOMS	NUMBER OF RESERVED ROOMS
PRIVATE	STRING	ROOM_TYPE	TYPE OF ROOMS
PRIVATE	STRING	AVAIL_NO_OF_ROOMS	NUMBER OF ROOMS AVAILABLE
PRIVATE	INTEGER	ROOM_RATE	ROOM RATE
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	RET_ROOM_DETAIL		
PUBLIC	SET_ROOM_TYPE		
PUBLIC	CAL_DISC_RATE		
PUBLIC	RET_AVAIL_ROOMS		
PUBLIC	UPDATE_RESERVE_DETAIL		

Table A-11 CLASS RESERVE_DETAIL

CLASS NAME	RESERVE_DETAIL	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS CONTAINS RESERVATION INFORMATION		
DATAMEMBERS/ INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	RESERVE_ID	RESERVATION ID
PRIVATE	STRING	CORP_ID	CORPORATE ID
PRIVATE	DATE	EXP_CHECK_IN	EXPECTED CHECK IN DATE
PRIVATE	DATE	EXP_CHECK_OUT	EXPECTED CHECK OUTDATE
PRIVATE	STRING	CUST_NAME	CUSTOMER NAME
PRIVATE	STRING	CUST_ADDR	CUSTOMER ADDRESS
PRIVATE	STRING	CUST_SEX	CUSTOMER SEX
PRIVATE	STRING	CUST_TEL	CUSTOMER TELEPHONE
PRIVATE	STRING	CUST_FAX	CUSTOMER FAX
PRIVATE	STRING	RES_STATE	STATE OF RESERVATION
PRIVATE	STRING	RESV_COLLECTION	ARRAY FOR RESERVATION COLLECTION
PRIVATE	STRING	REG_COLLECTIONS	ARRAY FOR REGISTRAION COLLECTION
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	RET_RESERVE_ID_LIST		
PUBLIC	RET_AVAIL_ROOMS		
PUBLIC	RET_DISCOUNT		
PUBLIC	RET_RESV_ID		
PUBLIC	RET_STATE		
PUBLIC	CANCEL_RESERVATION		

Table A-12 CLASS RESERVATION

CLASS NAME	ROOMS	DATE	1/5/1998
VERSION	1		
BASE CLASS			
BLOCK NAME			
AUTHOR	MR.CHUSAK S.		
ANALYSIS OBJECT			
CLASS PURPOSE /DESCRIPTION	THIS CLASS IS REFERENCE TABLE CONTAINING ROOMS INFORMATION		
DATAMEMBERS/ INFORMATION			
VISIBILITY	TYPE	NAME	DESCRIPTION
PRIVATE	STRING	ROOM_NO	ROOM NUMBER
PRIVATE	STRING	ROOM_DESC	ROOM DESCRIPTION
PRIVATE	STRING	ROOM_TYPE	(SINGLE, TWIN, SUITE)
PRIVATE	INTEGER	ROOM_RATE	ROOM RATE
PRIVATE	STRING	ROOM_STATE	ASSIGNED, EMPTY, REGISTERED, ASSIGNED_CHECK_IN
PRIVATE	STRING	WA_ROOM_LIST	WORK ARRAY ROOM LIST
MEMBER FUNCTIONS (FUNCTIONALITY)			
VISIBILITY	NAME	DESCRIPTION	
PUBLIC	CHANGE_STATE		
PUBLIC	RET_CHK_OUT_ROOM_LIST		
PUBLIC	RET_ROOM_STATE_LIST		
PUBLIC	CHK_VACANCY		
PUBLIC	RET_EMPTY_ROOM_LIST		

Table A-13 CLASS RESERVATION DETAIL

CLASS	BILL	DATE	1/5/1998	
FUNCTION	RET_TOT_MEAL_LAUNDRY_AMT			
PURPOSE	RETURN AVAILABLE TOTAL MEAL AND LAUNDRY AMOUNT			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	REGISTRATION_ID		REGISTRATION ID
OUT	INTEGER	TOTAL_MEAL		TOTAL MEAL
OUT	INTEGER	TOTAL_LAUNDRY		TOTAL LAUNDRY
DESCRIPTION/ SPECIFICATION				

Table A-14 FUNCTION RET_TOT_MEAL_LAUNDRY_AMT

CLASS	BILL	DATE	1/5/1998	
FUNCTION	CLOSE_BILL			
PURPOSE	CLOSE BILL AFTER PAYMENT			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	REGISTRATION_ID		REGISTRATION ID
IN	INTEGER	PAYMENT_OK		ALREADY PAID
IN	DATE	DATE		PAYMENT DATE
DESCRIPTION/ SPECIFICATION				

Table A-15 FUNCTION CLOSE_BILL

CLASS	BILL	DATE	1/5/1998	
FUNCTION	RET_RECEIPT_LIST			
PURPOSE	RETURN RECEIPT LIST TO ACCOUNT DEPARTMENT AT THE END OF THE DAY			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	WA_RECEIPT_LIST		REPORT OF RECEIPT LIST AT THE END OF THE DAY
DESCRIPTION/ SPECIFICATION				

Table 3.4:FUNCTION RET_RECEIPT_LIST

CLASS	CLOTHES	DATE	1/5/1998	
FUNCTION	RET_CLOTH_PRICE			
PURPOSE	RETURN CLOTHES PRICE			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	CLOTH_TYPE		GET CLOTHES TYPE
OUT	INTEGER	CLOTH_PRICE		RETURN CLOTH PRICE
DESCRIPTION/ SPECIFICATION				

Table A-16 FUNCTION RET_CLOTH_PRICE

CLASS	LAUNDRY	DATE	1/5/1998	
FUNCTION	CAL_LAUN_AMT			
PURPOSE	CALCULATE LAUNDRY AMOUNT AND UPDATE			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	DATE	LAUN_DATE		LAUNDRY DATE
IN	STRING	ROOM_NO		ROOM NUMBER
IN	STRING	CLOTH_TYPE		CLOTH TYPE
IN	STRING	CLOTH_QUANTITY		CLOTH QUANTITY
DESCRIPTION/ SPECIFICATION				

Table A-17 FUNCTION CAL_LAUN_AMT

CLASS	MEAL	DATE	1/5/1998	
FUNCTION	UPDATE_MEAL_AMT			
PURPOSE	UPDATE MEAL AMOUNT			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	DATE	MEAL_DATE		MEAL DATE
IN	STRING	ROOM_NO		ROOM NUMBER
IN	INTEGER	MEAL_AMOUNT		MEAL AMOUNT
DESCRIPTION/ SPECIFICATION				

Table A-18 FUNCTION UPDATE_MEAL_AMT

CLASS	PAYMENT	DATE	1/5/1998	
FUNCTION	CAL_TOT_AMT			
PURPOSE	CALCULATE TOTAL AMOUNT			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	REGISTRATION_ID		ROOM TYPE ARRAY STATE
IN	INTEGER	TOTAL_MEAL		
IN	INTEGER	TOTAL_LAUNDRY		
IN	INTEGER	LAST_MIN_CHARGE		
IN	INTEGER	LAST_MIN_CHARGE		
IN	DATE	ACT_CHECK_IN		DATE IN
IN	DATE	DATE		SYSTEM DATE
OUT	INTEGER	PAYMENT_AMOUNT		
DESCRIPTION/ SPECIFICATION				

Table A-19 FUNCTION CAL_TOT_AMT

CLASS	PAYMENT	DATE	1/5/1998	
FUNCTION	RET_PAY_TYPE_LIST			
PURPOSE	RETURN THE PAYMENT LIST AND THE DISCOUNT			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	PAYMENT_TYPE		PAYMENT TYPE
OUT	STRING	WA_PAYMENT_LIST		ARRAY
DESCRIPTION/ SPECIFICATION				

Table A-20 FUNCTION RET_PAY_TYPE_LIST

CLASS	PAYMENT	DATE	1/5/1998	
FUNCTION	CAL_NET_TOT_AMT			
PURPOSE	RETURN THE PAYMENT AMOUNT IF THE CUSTOMER USED CARD			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	W_CARD		
IN	INTEGER	DISCOUNT		
IN	INTEGER	TOTAL_AMT		
IN	STRING	CORP_ID		
OUT	STRING	PAYMENT_AMOUNT		
DESCRIPTION/ SPECIFICATION				

Table A-21 FUNCTION CAL_NET_TOT_AMT

CLASS	REGISTRATION	DATE	1/5/1998	
FUNCTION	RET_ACT_CHECK_OUT_LIST			
PURPOSE	RETURN LIST OF ROOMS FOR WHICH ACTUAL CHECK OUT DATE IS THE INPUT DATE			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	DATE	DATE	NULL	GET THE INPUT DATE
OUT	ARRAY OF STRING	WA_ROOMS	NULL	RETURN THE ROOM LIST
DESCRIPTION/ SPECIFICATION				

Table A-22 FUNCTION RET_ACT_CHECK_OUT_LIST

CLASS	REGISTRATION	DATE	1/5/1998	
FUNCTION	ASSIGN_ROOM_NO			
PURPOSE	TO ASSIGN ROOM NUMBERS TO A RESERVATION			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	ARRAY	A_ROOM_RO		GET ROOM NUMBER
IN	ARRAY	A_RESERVE_ID		GET RESERVATION ID
IN	ARRAY	A_STATE		GET CURRENT STATE OF THE ROOM
OUT	ARRAY OF STRING	WA_ROOMS	NULL	RETURN THE ROOM LIST
DESCRIPTION/ SPECIFICATION				

Table A-23 FUNCTION ASSIGN_ROOM_NO

CLASS	REGISTRATION	DATE	1/5/1998	
FUNCTION	RET_RESERVED_ROOM			
PURPOSE	RETURN RESERVED ROOM LIST FOR THE GIVEN RESERVE ID			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	RESERVE_ID		GET THE RESERVE ID
OUT	ARRAY	A_ROOMS		RETURN THE ROOM NUMBERS
DESCRIPTION/ SPECIFICATION				

Table A-24FUNCTION RET_RESERVED_ROOM

CLASS	REGISTRATION	DATE	1/5/1998	
FUNCTION	UPDATE_REGISTRATION			
PURPOSE	UPDATE REGISTRATION INTO DURING THE CHECK IN			
PARAMETER				
IN/OUT	TYPE	NAME	DEFAULT	PURPOSE
IN	STRING	CUST_NAME		CUSTOMER NAME
IN	STRING	PASSPORT_NO		PASSPORT NUMBER
IN	STRING	ID_CARD_NO		ID CARD NUMBER
IN	STRING	CONTACT_ADDRESS		CONTACT ADDRESS
IN	STRING	CONTACT_PHONE		CONTACT PHONE
IN	DATE	ACT_CHECK_OUT		ACTUAL CHECK OUT DATE
IN	STRING	ROOM_NO		ROOM NUMBER
IN	STRING	PAYMENT_TYPE		PAYMENT TYPE
OUT	STRING	REGISTRATION_ID		REGISTRATION ID

Table A-25 FUNCTION UPDATE_REGISTRATION

Database Design

Table A-26: RESERVATION TABLE

COLUMN NAME	TYPE	SIZE	KEY
RESERVE_ID	VARCHAR	10	X (P)
CARD_NAME	VARCHAR	10	
EXP_CHECK_IN	DATE		
EXP_CHECK_OUT	DATE		
CUST_NAME	VARCHAR	30	
CUST_ADDR	VARCHAR	50	
CUST_SEX	CHAR	1	
CUST_TEL	VARCHAR	12	
CUST_FAX	VARCHAR	12	
RES_STATE	INTEGER	1	

Table A-27 RESERVE_DETAIL TABLE

COLUMN NAME	TYPE	SIZE	KEY
RESERVE_ID	VARCHAR	10	X (P)
NO_OF_ROOMS	INTEGER	3	
ROOM_TYPE	INTEGER	1	

Table A-28 REGISTRATION TABLE

COLUMN NAME	TYPE	SIZE	KEY
REGISTRATION_ID	VARCHAR	13	X (P)
PERSON_NAME	VARCHAR	30	
RESERV_ID	VARCHAR	10	X (S)
PASSPORT_NO	VARCHAR	12	
ID_CARD_NO	VARCHAR	12	
CONTACT_ADDR	VARCHAR	50	
CONTACT_PHONE	VARCHAR	12	
ROOM_NO	VARCHAR	3	X (S)
ACT_CHECK_IN	DATE		
ACT_CHECK_OUT	DATE		
PAYMENT_TYPE	INTEGER	1	
REG_CLOSED	INTEGER	1	

Table A-29 LAUNDRY TABLE

COLUMN NAME	TYPE	SIZE	KEY
REGISTRATION_ID	VARCHAR	13	X (P)
LAUND_DATE	DATE		
LAUN_AMOUNT	REAL	5 69	
CLOTH_ID	VARCHAR	3	X (S)
CLOTH_QUANTITY	INTEGER	3	
ROOM_NO	VARCHAR	3	X (S)

Table A-30 COTHES TABLE

COLUMN NAME	TYPE	SIZE	KEY
CLOTH_ID	VARCHAR	3	X (P)
CLOTH_TYPE	VARCHAR	10	
CLOTH_PRICE	REAL	4	

Table A-31 PAYMENT TABLE

COLUMN NAME	TYPE	SIZE	KEY
PAYMENT_AMOUNT	REAL	6	
PAYMENT_TYPE	INTEGER	1	
CARD_NO	VARCHAR	15	
CARD_COMPANY	VARCHAR	10	
REGISTRATION_ID	VACHAR	13	X (P)
ROOM_NO	VARCHAR	3	X (S)
PAY_CLOSED	INTEGER	1	

Table A-32 ROOMS TABLE

COLUMN NAME	TYPE	SIZE	KEY
ROOM_NO	VARCHAR	3	X (P)
ROOM_DECS	VARCHAR	20	
ROOM_TYPE	INTEGER	1	
ROOM_RATE	REAL	5	
ROOM_STATE	INTEGER	1	

Table A-33 BILL TABLE

COLUMN NAME	TYPE	SIZE	KEY
REGISTRATION_ID	VARCHAR	13	X (P)
PAYMENT_TYPE	INTEGER	1	
TOTAL_MEAL	REAL	5	
TOTAL_LAUNDRY	REAL	5	
LAST_MIN_CHARGE	REAL	5	
SERVICE_CHARGE	REAL	5	
ROOM_NO	VARCHAR	3	X (S)
BILL_CLOSED	INTEGER	1	
TOTAL_AMOUNT	REAL	5	

Table A-34 MEAL TABLE

COLUMN NAME	TYPE	SIZE	KEY
REGISTRATION_ID	VARCHAR	13	X (P)
MEAL_DATE	DATE		
MEAL_AMOUNT	REAL	5	
MEAL_DESC	VARCHAR	20	
ROOM_NO	VARCHAR	3	X (S)

Table A-35 CORPORATE TABLE

COLUMN NAME	TYPE	SIZE	KEY
CORP_NAME	VARCHAR	10	X (P)
CORP_DISCOUNT	INTEGER	1	

Table A-36 CARDS TABLE

COLUMN NAME	TYPE	SIZE	KEY
CARD_COMPANY	VARCHAR	10	X (P)
CARD_TYPE	INTEGER	1	
DISCOUNT	REAL	3	

Appendix B

Screen Code

Toc.html

```
<html>

<body background=http://cicc01.serve/gif/backg1.gif>

<center><h3>Table of Contents</h3></center>
<hr>
<A href=intro.htm target=diagram>Introduction</a><p>
<A href=diag.htm target=diagram>Class Definitions</a><p>
<A href=workf.htm target=diagram>Work Flow</a>
Interaction Diagrams for Scenarios:
<ul>
<li><A href=doc/scn_res.htm target=diagram>Reservation</A>
<li>Check out list preparation (see assigning)
<li><A href=doc/scn_asg.htm target=diagram>Assigning</A>
<li><A href=doc/scn_cin.htm target=diagram>Check-in</A>
<li>Registered list(see Check-in)
<li><A href=doc/scn_out.htm target=diagram>Check-out</A>
<li><A href=doc/scn_mnl.htm target=diagram>Meal and Laundry</A>
<li><A href=doc/scn_ext.htm target=diagram>Extending</A>
<li><A href=doc/scn_cnl.htm target=diagram>Cancel</A>
</ul>
<A href=prot.htm target=diagram>Prototype</a><p>
</body>
</html>
```

Reservation.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>
<center>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>RESERVATION - Check available rooms</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg" bgcolor="#FFFFFF">
```

```

<h1 align="left"><font color="#FF0080" size="4">RESERVATION-Choose
the stay period and rooms</font></h1>
<!--webbot bot="GeneratedScript" preview="" startspan --><script
language="JavaScript"><!--
function FrontPage_Form1_Validator(theForm)
{

    if (theForm.TXCHECKINDAY.value == "")
    {
        alert("Please enter a value for the 'TXCHECKINDAY' field.");
        theForm.TXCHECKINDAY.focus();
        return (false);
    }

    if (theForm.TXCHECKINDAY.value.length < 1)
    {
        alert("Please enter at least 1 characters in the 'TXCHECKINDAY' field.");
        theForm.TXCHECKINDAY.focus();
        return (false);
    }

    if (theForm.TXCHECKINDAY.value.length > 2)
    {
        alert("Please enter at most 2 characters in the 'TXCHECKINDAY' field.");
        theForm.TXCHECKINDAY.focus();
        return (false);
    }

    var checkOK = "0123456789-";
    var checkStr = theForm.TXCHECKINDAY.value;
    var allValid = true;
    var decPoints = 0;
    var allNum = "";
    for (i = 0; i < checkStr.length; i++)
    {
        ch = checkStr.charAt(i);
        for (j = 0; j < checkOK.length; j++)
            if (ch == checkOK.charAt(j))
                break;
        if (j == checkOK.length)
        {
            allValid = false;
            break;
        }
        allNum += ch;
    }
    if (!allValid)
    {
        alert("Please enter only digit characters in the 'TXCHECKINDAY' field.");
        theForm.TXCHECKINDAY.focus();
    }
}

```

```

return (false);
}

if (theForm.TXCHECKOUTMONTH.value == "")
{
    alert("Please enter a value for the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

if (theForm.TXCHECKOUTMONTH.value.length < 1)
{
    alert("Please enter at least 1 characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

if (theForm.TXCHECKOUTMONTH.value.length > 2)
{
    alert("Please enter at most 2 characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXCHECKOUTMONTH.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}
if (!allValid)
{
    alert("Please enter only digit characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}
return (true);
}

```

```

--></script><!--webbot bot="GeneratedScript" endspan -->
<form action="/hotel/scripts/reservation.pl"
method="POST" onsubmit="return FrontPage_Form1_Validator(this)"
name="FrontPage_Form1">
  <input type="hidden" name="HDCHECKIN" value="1"><input
type="hidden" name="HDCHECKOUT" value="1"><div align="center"><center><table
border="1">
  <tr>
    <td><font size="1"><strong>Check in</strong></font></td>
    <td><font size="1"><strong><input type="text"
size="4" name="TXCHECKINYEAR">YYYY</strong></font></td>
    <td><font size="1"><strong><input type="text"
size="2" name="TXCHECKINMONTH">MM</strong></font></td>
    <td><font size="1"><strong><!--webbot
bot="Validation" s-data-type="Integer"
s-number-separators="x" b-value-required="TRUE"
i-minimum-length="1" i-maximum-length="2" --><input
type="text" size="2" maxlength="2"
name="TXCHECKINDAY">DD</strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>Check out</strong></font></td>
    <td><font size="1"><strong><input type="text"
size="4" name="TXCHECKOUTYEAR">YYYY</strong></font></td>
    <td><font size="1"><strong><!--webbot
bot="Validation" s-data-type="Integer"
s-number-separators="x" b-value-required="TRUE"
i-minimum-length="1" i-maximum-length="2" --><input
type="text" size="2" maxlength="2"
name="TXCHECKOUTMONTH">MM</strong></font></td>
    <td><font size="1"><strong><input type="text"
size="2" name="TXCHECKOUTDAY">DD</strong></font></td>
  </tr>
  <tr>
    <td valign="top"><font size="1"><strong>Type of rooms</strong></font></td>
    <td colspan="3" valign="top"><font size="1"><strong>Number
of rooms</strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>Single</strong></font></td>
    <td colspan="3"><font size="1"><strong><input
type="text" size="2" name="TXSINGLE" value="0"></strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>Twin</strong></font></td>
    <td colspan="3"><font size="1"><strong><input
type="text" size="2" name="TXTWIN" value="0"></strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>Suite</strong></font></td>

```

```

        <td colspan="3"><font size="1"><strong><input
        type="text" size="2" name="TXSUITE" value="0"></strong></font></td>
    </tr>
</table>
</center></div><p align="center"><input type="submit"
name="BTSSUBMIT" value="SUBMIT"> <input type="button"
name="BTRESET" value="RESET"></p>
<p align="center"><script language="JavaScript"><!--
function FrontPage_Form1_Validator(theForm)
{

if (theForm.TXCHECKINYEAR.value == "")
{
    alert("Please enter a value for the \"TXCHECKINYEAR\" field.");
    theForm.TXCHECKINYEAR.focus();
    return (false);
}

if (theForm.TXCHECKINYEAR.value.length < 4)
{
    alert("Please enter at least 4 characters in the \"TXCHECKINYEAR\" field.");
    theForm.TXCHECKINYEAR.focus();
    return (false);
}

if (theForm.TXCHECKINYEAR.value.length > 4)
{
    alert("Please enter at most 4 characters in the \"TXCHECKINYEAR\" field.");
    theForm.TXCHECKINYEAR.focus();
    return (false);
}

var checkOK = "0123456789-.";
var checkStr = theForm.TXCHECKINYEAR.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    if (ch != ".")
        allNum += ch;
}
}
}

```

```

}
if (!allValid)
{
    alert("Please enter only digit characters in the 'TXCHECKINYEAR' field.");
    theForm.TXCHECKINYEAR.focus();
    return (false);
}

if (theForm.TXCHECKINMONTH.value == "")
{
    alert("Please enter a value for the 'TXCHECKINMONTH' field.");
    theForm.TXCHECKINMONTH.focus();
    return (false);
}

if (theForm.TXCHECKINMONTH.value.length < 1)
{
    alert("Please enter at least 1 characters in the 'TXCHECKINMONTH' field.");
    theForm.TXCHECKINMONTH.focus();
    return (false);
}

if (theForm.TXCHECKINMONTH.value.length > 2)
{
    alert("Please enter at most 2 characters in the 'TXCHECKINMONTH' field.");
    theForm.TXCHECKINMONTH.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXCHECKINMONTH.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
    {
        if (ch == checkOK.charAt(j))
            break;
    }
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}
if (!allValid)
{
    alert("Please enter only digit characters in the 'TXCHECKINMONTH' field.");
}

```

```

theForm.TXCHECKINMONTH.focus();
return (false);
}

if (theForm.TXCHECKINDAY.value == "")
{
    alert("Please enter a value for the \"TXCHECKINDAY\" field.");
    theForm.TXCHECKINDAY.focus();
    return (false);
}

if (theForm.TXCHECKINDAY.value.length < 1)
{
    alert("Please enter at least 1 characters in the \"TXCHECKINDAY\" field.");
    theForm.TXCHECKINDAY.focus();
    return (false);
}

if (theForm.TXCHECKINDAY.value.length > 2)
{
    alert("Please enter at most 2 characters in the \"TXCHECKINDAY\" field.");
    theForm.TXCHECKINDAY.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXCHECKINDAY.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}
if (!allValid)
{
    alert("Please enter only digit characters in the \"TXCHECKINDAY\" field.");
    theForm.TXCHECKINDAY.focus();
    return (false);
}

```

```

if (theForm.TXCHECKOUTYEAR.value == "")
{
    alert("Please enter a value for the 'TXCHECKOUTYEAR' field.");
    theForm.TXCHECKOUTYEAR.focus();
    return (false);
}

if (theForm.TXCHECKOUTYEAR.value.length < 4)
{
    alert("Please enter at least 4 characters in the 'TXCHECKOUTYEAR' field.");
    theForm.TXCHECKOUTYEAR.focus();
    return (false);
}

if (theForm.TXCHECKOUTYEAR.value.length > 4)
{
    alert("Please enter at most 4 characters in the 'TXCHECKOUTYEAR' field.");
    theForm.TXCHECKOUTYEAR.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXCHECKOUTYEAR.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
    {
        if (ch == checkOK.charAt(j))
        {
            break;
        }
        if (j == checkOK.length)
        {
            allValid = false;
            break;
        }
    }
    allNum += ch;
}

if (!allValid)
{
    alert("Please enter only digit characters in the 'TXCHECKOUTYEAR' field.");
    theForm.TXCHECKOUTYEAR.focus();
    return (false);
}

if (theForm.TXCHECKOUTMONTH.value == "")
{
    alert("Please enter a value for the 'TXCHECKOUTMONTH' field.");
    theForm.TXCHECKOUTMONTH.focus();
}

```

```

return (false);
}

if (theForm.TXCHECKOUTMONTH.value.length < 1)
{
    alert("Please enter at least 1 characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

if (theForm.TXCHECKOUTMONTH.value.length > 2)
{
    alert("Please enter at most 2 characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXCHECKOUTMONTH.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
    {
        if (ch == checkOK.charAt(j))
        {
            break;
        }
        if (j == checkOK.length)
        {
            allValid = false;
            break;
        }
    }
    allNum += ch;
}

if (!allValid)
{
    alert("Please enter only digit characters in the \"TXCHECKOUTMONTH\" field.");
    theForm.TXCHECKOUTMONTH.focus();
    return (false);
}

if (theForm.TXSINGLE.value.length > 2)
{
    alert("Please enter at most 2 characters in the \"TXSINGLE\" field.");
    theForm.TXSINGLE.focus();
    return (false);
}

```

```

var checkOK = "0123456789-";
var checkStr = theForm.TXSINGLE.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}
if (!allValid)
{
    alert("Please enter only digit characters in the 'TXSINGLE' field.");
    theForm.TXSINGLE.focus();
    return (false);
}

if (theForm.TXTWIN.value.length > 2)
{
    alert("Please enter at most 2 characters in the 'TXTWIN' field.");
    theForm.TXTWIN.focus();
    return (false);
}

var checkOK = "0123456789-";
var checkStr = theForm.TXTWIN.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}

```

```

}
if (!allValid)
{
    alert("Please enter only digit characters in the 'TXTWIN' field.");
    theForm.TXTWIN.focus();
    return (false);
}

```

```

if (theForm.TXSUITE.value.length > 1)
{
    alert("Please enter at most 1 characters in the 'TXSUITE' field.");
    theForm.TXSUITE.focus();
    return (false);
}

```

```

var checkOK = "0123456789-";
var checkStr = theForm.TXSUITE.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
    ch = checkStr.charAt(i);
    for (j = 0; j < checkOK.length; j++)
        if (ch == checkOK.charAt(j))
            break;
    if (j == checkOK.length)
    {
        allValid = false;
        break;
    }
    allNum += ch;
}
if (!allValid)
{
    alert("Please enter only digit characters in the 'TXSUITE' field.");
    theForm.TXSUITE.focus();
    return (false);
}

```

```

// own code
var winyy = theForm.TXCHECKINYEAR.value;
var winmm = theForm.TXCHECKINMONTH.value ;
var windd = theForm.TXCHECKINDAY.value;

var Today = new Date();

```

```

var woutyy = Today.getYear();
var woutmm = Today.getMonth();
var woutdd = Today.getDate();
woutmm = eval(woutmm + 1);
var stat;
var leapin, leap1;
leap = winyy.charAt(2);
leap += winyy.charAt(3);

stat = wcheck(leap, winmm, windd, woutyy, woutmm, woutdd, 1);
// alert(stat);
if ( !stat )
{
    alert("Check in is before today's date");
    return (false);
}

woutyy = theForm.TXCHECKOUTYEAR.value;
woutmm = theForm.TXCHECKOUTMONTH.value ;
woutdd = theForm.TXCHECKOUTDAY.value;

leap1 = woutyy.charAt(2);
leap1 += woutyy.charAt(3);

stat = wcheck(leap, winmm, windd, leap1, woutmm, woutdd, 0);

if ( !stat )
{
    alert("error");
    return (false);
}

theForm.HDCHECKIN.value = ( winyy + "/" + winmm + "/" + windd );
theForm.HDCHECKOUT.value = ( woutyy + "/" + woutmm + "/" + woutdd );

if (( theForm.TXSINGLE.value == "0" ) && ( theForm.TXTWIN.value == "0" ) && ( theForm.TXSUITE.value == "0" ) )
{
    alert("Please select a room");
    return(false);
}

if (( theForm.TXSINGLE.value == "0" ) && ( theForm.TXTWIN.value == "0" ) && ( theForm.TXSUITE.value == "0" ) )
{
    alert("Please select a room");
    return(false);
}

return (true);

}

function wcheck(winyy, winmm, windd, woutyy, woutmm, woutdd, flag)
{
    var dayarray = new Array();
    dayarray[0] = 28;

```

```

dayarray[1] = 31;
dayarray[2] = 28;
dayarray[3] = 31;
dayarray[4] = 30;
dayarray[5] = 31;
dayarray[6] = 30;
dayarray[7] = 31;
dayarray[8] = 31;
dayarray[9] = 30;
dayarray[10] = 31;
dayarray[11] = 30;
dayarray[12] = 31;
// alert (winyy);
var leap = winyy;
// alert(leap);
if ( ( leap % 4 ) == 0 )
{
    dayarray[2] = 29;
}
var leap = woutyy;
if ( ( leap % 4 ) == 0 )
{
    dayarray[0] = 29;
}

var indd;
var outdd;
indd = ( winyy + "/" + winmm + "/" + windd );
alert(indd);
if ( windd > dayarray[winmm] )
{
    alert( "Number of days is incorrect in in date");
    return (false);
}

outdd = ( woutyy + "/" + woutmm + "/" + woutdd );
// alert( outdd);

dayarray[2] = dayarray[0];

if ( woutdd > dayarray[woutmm] )
{
    alert( "Number of days is incorrect in out date");
    return (false);
}

var current;

current = new Date();

```

```

// check year
// alert(" check year ");
if ( winyy < woutyy )
{
    return (true);
}
else
{
    if ( winyy > woutyy )
    {
        alert( "Check in year is after check out year");
        return ( false);
    }
}
// Check month
// alert (" check month ");
if ( winmm > woutmm )
{
    alert( "Check in month is after check out month");
    return ( false);
}
else
{
    if ( woutmm > winmm )
    {
        return (true);
    }
}
// Check day
// alert (" check day ");
// alert(windd);
// alert(woutdd);
var owindd = eval(windd);
var owoutdd = eval(woutdd);

if ( flag != 1 )
(
    if ( ( owindd > owoutdd ) || ( windd == woutdd ) )
    {
        alert( "Check in day is after or same as check out day");
        return ( false);
    }
}
else
{
    if ( windd < woutdd )
    {
        alert( "In date is before today");
        return ( false);
    }
}

```

```

    }
}
return (true);

}
// --></script></p>

</form>
<br><a href=prot.htm>Back</a>
</center>
</body>
</html>

```

Reservation_ok.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>RESERVATION OK!- Update finish..</title>
</head>

<body bgcolor="white">
<center>
<h1 align="left"><font color="#FF0080" size="4">RESERVATION OK! -
Reserved the rooms done</font></h1>

<form action="http://cicc09/hotel/scripts/reservation.pl"
method="POST" target="F1"
onsubmit="return FrontPage_Form1_Validator(this)">
  <input type="hidden" name="HDADDRESS" value="1"><input
type="hidden" name="HDCHECKIN" value="1"><input type="hidden"
name="HDCHECKOUT" value="1"><input type="hidden"
name="HDCORP_NAME" value="1"><input type="hidden"
name="HDNAME" value="1"><input type="hidden" name="HDPHONE"
value="1"><input type="hidden" name="HDSEX" value="1"><input
type="hidden" name="HDSINGLE" value="1"><input type="hidden"
name="HDSUITE" value="1"><input type="hidden" name="HDTWIN"
value="1"><div align="center"><center><table border="1">
  <tr>
    <td><font size="1"><strong>EXPECTED DATE IN</strong></font></td>
    <td><font size="1"><strong><input type="text"
size="14" name="TXEXPIN"></strong></font></td>
  </tr>
</tr>

```

```

        <td><font size="1"><strong>EXPECTED DATE OUT</strong></font></td>
        <td><font size="1"><strong><input type="text"
size="14" name="TXEXPOUT"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>Single</strong></font></td>
        <td colspan="3"><font size="1"><strong><input
type="text" size="2" name="TXSINGLE" value="0"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>Twin</strong></font></td>
        <td colspan="3"><font size="1"><strong><input
type="text" size="2" name="TXTWIN" value="0"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>Suite</strong></font></td>
        <td colspan="3"><font size="1"><strong><input
type="text" size="2" name="TXSUITE" value="0"></strong></font></td>
    </tr>
</table>
</center></div><table border="1">
    <tr>
        <td><font size="1"><strong>NAME :</strong></font></td>
        <td><font size="1"><strong><input type="text"
size="20" name="TXCUST_NAME" 30"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>ADDRESS :</strong></font></td>
        <td><font size="1"><strong><input type="text"
size="40" name="TXCUST_ADDRESS"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>PHONE</strong></font></td>
        <td><font size="1"><strong><input type="text"
size="15" name="TXCUST_TEL"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>SEX</strong></font></td>
        <td><font size="1"><strong>MALE</strong><input
type="checkbox" name="C1"> FEMALE<input type="radio"
checked name="R1" value="V1"></strong></font></td>
    </tr>
    <tr>
        <td><font size="1"><strong>CORP NAME</strong></font></td>
        <td><font size="2"><input type="text" size="20"
name="TXCORP_NAME"></font></td>
    </tr>
</table>
</form>

```

```
<a href=prot.htm>Back</a>
</center>
</body>
</html>
```

Reservation_error.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>RESERVATION-error</title>
</head>

<body bgcolor="#FFFFFF">

<p><font color="#FF0080" size="5">RESERVATION-Reserve room grater
than available rooms</font></p>

<p>&nbsp;</p>

<hr>

<p align="center"><a href="reservation.htm">Back to Reservation</a></p>
</body>
</html>
```

Register.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Register-Ask reserve id</title>
</head>

<body background="bg080.jpg">

<h1>Register-Ask reserve ID </h1>

<p><font size="4"><strong>Ask reserve id for check in information</strong></font></p>
```

<hr>

```
<form action="http://cicc09/hotel/scripts/regis_in.pl"
method="POST" name="f1">
  <div align="center"><center><table border="1">
    <tr>
      <td><strong>Reserve ID</strong> : </td>
      <td><input type="text" size="20" name="T1"></td>
    </tr>
  </table>
  </center></div><p align="center"><input type="submit"
name="BTSUBMIT" value="Submit"><input type="reset"
name="BTRESET" value="Reset"> </p>
</form>
```

<p> </p>

<hr>

Back

</body>

</html>

Reservation_info.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Registration-Get customer information</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080">Registration-Get customer information</font></h1>
<div align="center"><center>

<table border="1">
  <tr>
    <td><font size="2"><strong>RESERVE ID : </strong></font></td>
    <td><font size="2"><strong><input type="text" size="12"
name="TXRESERVE_ID" value="1234567890"></strong></font></td>
    <td><font size="2"><strong>REGISTER ID :</strong></font></td>
    <td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890101"> </strong></font></td>
    <td><font size="2"><strong>CHECK IN :<input type="text"
size="15" name="TXDATE" value="1997/12/18"></strong></font></td>
```

```

</tr>
</table>
</center></div>

<form method="POST" action="_vti_bin/shtml.dll/regis_info.htm"
webbot-action="--WEBBOT-SELF--">
  <!--webbot bot="SaveResults" u-file="formrsit.htm"
s-format="HTML/DL" b-label-fields="TRUE" startspan --><input
type="hidden" name="VTI-GROUP" value="0"><!--webbot
bot="SaveResults" endspan i-checksum="47042" --><div
align="center"><center><table border="0">
  <tr>
    <td><font size="1">ROOM NO.</font></td>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1">NAME :</font></td>
    <td><font size="1"><input type="text" size="30"
name="TXPERSON_NAME" value="MR.XXXXXXXXXXXXXXX"></font></td>
  </tr>
  <tr>
    <td><font size="1"><input type="text" size="10"
name="TXROOM_NO" value="101"></font></td>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1">PASSPORT NO.</font></td>
    <td><font size="1"><input type="text" size="20"
name="TXPASSPORT_NO" value="K12312345"></font></td>
  </tr>
  <tr>
    <td><font size="1">TYPE</font></td>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1">ID NO.</font></td>
    <td><font size="1"><input type="text" size="20"
name="TXID_CARD_NO" value="Z123456789"></font></td>
  </tr>
  <tr>
    <td><font size="1"><input type="text" size="10"
name="TXROOM_TYPE" value="SINGLE"></font></td>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1">ADDRESS</font></td>
    <td><font size="1"><input type="text" size="40"
name="TXCONTACT_ADDRESS"
value="123 YOKOHAMA KENSHU CENTER, YOKOHAMA"></font></td>
  </tr>
  <tr>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1"></font>&nbsp;</td>
    <td><font size="1">PHONE</font></td>
    <td><font size="1"><input type="text" size="20"
name="TX_CONTACT_PHONE" value="054 1234567"></font></td>
  </tr>
  <tr>

```

```

<td><font size="1"></font>&nbsp;</td>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1">CHECK OUT</font></td>
<td><font size="1"><input type="text" size="20"
name="TXACT_CHECK_OUT" value="1997/12/25"></font></td>
</tr>
<tr>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1">PAYMENT</font></td>
<td><font size="1"><select name="PAYMENT_TYPE"
size="1">
<option selected value="CASH">CASH</option>
<option value="COMPANY">COMPANY</option>
<option value="CREDIT CARD">CREDIT CARD</option>
<option value="TRAVEL CHECK">TRAVEL CHECK</option>
</select></font></td>
</tr>
</table>
</center></div><p align="center"><input type="submit"
value="Submit Form"> <input type="reset" value="Reset Form"> </p>
</form>
<hr>
<h5 align="center">Copyright information goes here,<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->March 10, 1998<!--webbot
bot="TimeStamp" endspan i-checksum="41163" --></h5>
</body>
</html>

```

Assign.html

```

<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>ASSIGN-Assign the room</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080">ASSIGN-Get the reserve ID by date</font></h1>

```

```

<hr>

<p>
<script language="VBScript">
<sub window_onload
    F1.txyear.value=year(date)
    F1.txmonth.value=month(date)
    F1.txday.value=day(date)
    F1.txday.focus
end sub>
</script> </p>
<center>
<form action="/hotel/scripts/getres.pl" method="POST" name="f1">
    <input type="hidden" name="HDDATE" value="1"><div
    align="center"><center><table border="1">
        <tr>
            <td>ASSIGN DATE</td>
            <td><input type="text" size="4" name="TXYEAR"> YYYY </td>
            <td><input type="text" size="2" name="TXMONTH"> MM </td>
            <td><input type="text" size="2" name="TXDAY"> DD </td>
        </tr>
    </table>
    </center></div><p align="center"><script language="VBScript"><!--sub btassign onclickf1.hddate.value=f1.txyear.value & "/" &
f1.txmonth.value & "/" & f1.txday.value msgbox f1.hddate.valueend sub--> </script>
    </p>
    <p align="center"><input type="submit" name="BTASSIGN"
    value="ASSIGN"> </p>
</form>

<br><a href=prot.htm>Back</a><br>
<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->March 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</center>
</body>
</html>

```

Assign room number.html

```

<html>
<title>Assign room number</title>
<body background="bg080.jpg">
<center>

```

```

<font size="1">
<table border="1">
  <tr>
    <td>RESERVATION ID : 1234567890 </td>
    <td>CHECK IN DATE : 1997/12/16 </td>
    <td>CHECK OUT DATE : 1997/12/25</td>
  </tr>
</table>
<table border="1">
  <tr>
    <td>ROOM TYPE</td>
    <td>SINGLE</td>
    <td>NO.OF ROOMS</td>
    <td> 2</td>
  </tr>
  <tr>
    <td>ROOM TYPE</td>
    <td>TWIN</td>
    <td>NO. OF ROOMS</td>
    <td>1</td>
  </tr>
  <tr>
    <td>ROOM TYPE</td>
    <td>SUITE</td>
    <td>NO. OF ROOMS</td>
    <td>0</td>
  </tr>
</table>
<form action="XX.PL" method="POST">
<table border="1">
  <tr>
    <td>ROOM NO</td>
    <td>ROOM TYPE</td>
    <td>ROOM STATE</td>
    <td>CHOOSED</td>
  </tr>
  <tr>
    <td>101</td>
    <td>SINGLE</td>
    <td>EMPTY</td>
    <td><input type="checkbox" name="C1"></td>
  </tr>
  <tr>
    <td>201</td>
    <td>SINGLE</td>
    <td>EMPTY</td>
    <td><input type="checkbox" name="C6"></td>
  </tr>
  <tr>
    <td>225</td>

```

```

        <td>SINGLE</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C7"></td>
    </tr>
    <tr>
        <td>250</td>
        <td>TWIN</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C8"></td>
    </tr>
    <tr>
        <td>223</td>
        <td>SINGLE</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C2"></td>
    </tr>
    <tr>
        <td>131</td>
        <td>TWIN</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C3"></td>
    </tr>
    <tr>
        <td>301</td>
        <td>SUITE</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C4"></td>
    </tr>
    <tr>
        <td>303</td>
        <td>SUITE</td>
        <td>EMPTY</td>
        <td><input type="checkbox" name="C5"></td>
    </tr>
</table>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</center>
</form>
</body>
</html>

```

Cancel.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>

```

```

<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>CANCEL-Cancel reservation</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080">CANCEL-Cancel reservation</font></h1>

<p>Ask reservation ID for cancel it.</p>

<form action="/hotel/scripts/cancel_re.pl"
method="POST" name="F1">
  <div align="center"><center><table border="1">
    <tr>
      <td><strong>RESERVATION ID </strong>:</td>
      <td><input type="text" size="20" name="TXRESERVE_ID"></td>
    </tr>
  </table>
</center></div><p align="center">&nbsp;</p>
<p align="center"><input type="submit" name="BTSUBMIT"
value="Submit "> <input type="reset" name="RESET"
value="Reset "> </p>
</form>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</body>
</html>

```

Check_out_end_of_day.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>CHECK OUT END OF DAY- Check out at the end of day</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

```

```

<body background="bg080.jpg">

<h1><font color="#FF0080" size="3">CHECK OUT END OF DAY-Check out
report at the end of day<script language="VBScript"><!--
sub window_onload
f1.txdate.value=YEAR(date)
f1.txdate.value=f1.txdate.value & "/" & MONTH(date)
f1.txdate.value=f1.txdate.value & "/" & DAY(date) +1
end sub
--></script></font></h1>

<form action="/hotel/scripts/exp_chkot.pl"
method="POST" name="F1" onsubmit>
  <div align="center"><center><table border="1">
    <tr>
      <td><strong>CHECK OUT DATE</strong></td>
      <td><font size="1"><input type="text" size="10"
name="TXDATE"><strong>YYYY/MM/DD</strong></font></td>
    </tr>
  </table>
</center></div><p align="center"><input type="submit"
name="BTSUBMIT" value="Submit"> <input type="reset"
value="Reset Form"> </p>
</form>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 16, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41187" --></h5>
</body>
</html>

```

Check_out_report.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>CHECK OUT REPORT- Check out report</title>
</head>

<body background="bg080.jpg" bgcolor="#FFFFFF">

```

<p>CHECK OUT REPORT- Check
out report at the end of the day</p>

<p>DATE : <input type="text" size="20" name="T1"></p>

<table border="1">

<tr>

<td>REGISTER ID</td>

<td>ROOM TYPE</td>

<td>ROOM NO.</td>

<td>CUSTOMER NAME</td>

<td>SEX</td>

<td>CHECK IN</td>

<td>CHECK OUT</td>

<td>COMPANY</td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>



[illegible]

```

<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>

<p align="center"><font size="1"></font></p>
<a href=prot.htm>Back</a>
</body>
</html>

```

Check_out_detail.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>CHECK OUT - Check out detail</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="5">CHECK OUT - Check out detail</font></h1>

<form action="_vti_bin/shtml.dll/chkout_d.htm" method="POST"
webbot-action="—WEBBOT-SELF—">
  <!--webbot bot="SaveResults" u-file="formrsit.htm"
s-format="HTML/DL" b-label-fields="TRUE" startspan --><input
type="hidden" name="VTI-GROUP" value="0"><!--webbot
bot="SaveResults" endspan i-checksum="47042" --><div

```

align="center"><center><table border="1" width="80%">

<tr>

<td>REGISTER ID: </td>

<td><input type="text" size="20"
name="T1"></td>

<td>CHECK IN DATE : </td>

<td><input type="text" size="20"
name="T2"></td>

</tr>

<tr>

<td>ROOM NO. </td>

<td><input type="text" size="20"
name="T3"></td>

<td>CHECK OUT DATE : </td>

<td><input type="text" size="20"
name="T4"></td>

</tr>

<tr>

<td>ROOM TYPE</td>

<td><input type="text" size="20"
name="T5"></td>

<td>PAYMENT TYPE</td>

<td><input type="text" size="20" name="T15"></td>

</tr>

<tr>

<td>NAME : </td>

<td colspan="3"><input type="text"
size="45" name="T6"></td>

</tr>

</table>

</center></div><div align="center"><center><table border="1">

<tr>

<td>ROOM CHARGE</td>

<td><input type="text" size="20"
name="T7"></td>

</tr>

<tr>

<td>TOTAL MEAL</td>

<td><input type="text" size="20"
name="T8"></td>

</tr>

<tr>

<td>TOTAL LAUNDRY</td>

<td><input type="text" size="20"
name="T9"></td>

</tr>

<tr>

<td>LAST MINUIE CHARGE</td>

<td><input type="text" size="20"
name="T10"></td>

```

</tr>
<tr>
  <td><font size="1">SERVICE CHARGE</font></td>
  <td><font size="1"><input type="text" size="20"
    name="T11"></font></td>
</tr>
<tr>
  <td><font size="1">TAX</font></td>
  <td><font size="1"><input type="text" size="20"
    name="T12"></font></td>
</tr>
<tr>
  <td><font size="1">DISCOUNT</font></td>
  <td><font size="1"><input type="text" size="20"
    name="T13"></font></td>
</tr>
<tr>
  <td><font size="1">TOTAL</font></td>
  <td><font size="1"><input type="text" size="20"
    name="T14"></font></td>
</tr>
</table>
</center></div><p align="center"><input type="submit"
name="BTSUBMIT" value="Submit "> <input type="reset"
value="Reset Form"> </p>
</form>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</body>
</html>

```

Checkout_get_detail.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>CHECK OUT- Check out get room no.</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

```

```
<body background="bg080.jpg">
```

```
<h1><font color="#FF0080" size="4">CHECK OUT - Check out room  
number</font></h1>
```

```
<hr>
```

```
<form action="/hotel/scripts/chkout_d.p"
```

```
method="POST" name="F1" onsubmit>
```

```
<p align="center">&nbsp;</p>
```

```
<div align="center"><center><table border="1">
```

```
<tr>
```

```
<td><font size="1"><strong>ROOM NO. </strong></font></td>
```

```
<td><select name="TXROOM_NO" size="1">
```

```
<option value="101">101</option>
```

```
<option value="102">102</option>
```

```
<option value="103">103</option>
```

```
<option value="104">104</option>
```

```
<option value="105">105</option>
```

```
<option value="106">106</option>
```

```
<option value="107">107</option>
```

```
<option value="108">108</option>
```

```
<option value="109">109</option>
```

```
<option value="110">110</option>
```

```
<option value="111">111</option>
```

```
<option value="112">112</option>
```

```
<option value="113">113</option>
```

```
<option value="114">114</option>
```

```
<option value="115">115</option>
```

```
<option value="116">116</option>
```

```
<option value="117">117</option>
```

```
<option value="118">118</option>
```

```
<option value="119">119</option>
```

```
<option value="120">120</option>
```

```
<option value="121">121</option>
```

```
<option value="122">122</option>
```

```
<option value="123">123</option>
```

```
<option value="124">124</option>
```

```
<option value="125">125</option>
```

```
<option value="126">126</option>
```

```
<option value="127">127</option>
```

```
<option value="128">128</option>
```

```
<option value="129">129</option>
```

```
<option value="130">130</option>
```

```
<option value="131">131</option>
```

```
<option value="132">132</option>
```

```
<option value="133">133</option>
```

```
<option value="134">134</option>
```

```
<option value="135">135</option>
```

<option value="136">136</option>
<option value="137">137</option>
<option value="138">138</option>
<option value="139">139</option>
<option value="140">140</option>
<option value="141">141</option>
<option value="142">142</option>
<option value="143">143</option>
<option value="144">144</option>
<option value="145">145</option>
<option value="201">201</option>
<option value="202">202</option>
<option value="203">203</option>
<option value="204">204</option>
<option value="205">205</option>
<option value="206">206</option>
<option value="207">207</option>
<option value="208">208</option>
<option value="209">209</option>
<option value="210">210</option>
<option value="211">211</option>
<option value="212">212</option>
<option value="213">213</option>
<option value="214">214</option>
<option value="215">215</option>
<option value="216">216</option>
<option value="217">217</option>
<option value="218">218</option>
<option value="219">219</option>
<option value="220">220</option>
<option value="221">221</option>
<option value="222">222</option>
<option value="223">223</option>
<option value="224">224</option>
<option value="225">225</option>
<option value="226">226</option>
<option value="227">227</option>
<option value="228">228</option>
<option value="229">229</option>
<option value="230">230</option>
<option value="231">231</option>
<option value="232">232</option>
<option value="233">233</option>
<option value="234">234</option>
<option value="235">235</option>
<option value="236">236</option>
<option value="237">237</option>
<option value="238">238</option>
<option value="239">239</option>
<option value="240">240</option>



```

<option value="241">241</option>
<option value="242">242</option>
<option value="243">243</option>
<option value="244">244</option>
<option value="245">245</option>
<option value="246">246</option>
<option value="247">247</option>
<option value="248">248</option>
<option value="249">249</option>
<option value="250">250</option>
<option value="251">251</option>
<option value="252">252</option>
<option value="301">301</option>
<option value="302">302</option>
<option value="303">303</option>
</select></td>
<td><font size="1"><strong>CHECK OUT DATE </strong></font></td>
<td><font size="1"><input type="text" size="15"
name="TXACT_CHECK_OUT"><strong> YYYY/MM/DD</strong></font></td>
</tr>
<tr>
<td><font size="2"><strong>LAST MIN CHARGE</strong></font></td>
<td><font size="2"><input type="text" size="14"
name="LAST_MIN" value="0"></font></td>
</tr>
</table>
</center></div><p align="center">&nbsp;</p>
<p align="center"><script language="VBScript"><!--
sub window_onload
f1.txACT_CHECK_OUT.value=YEAR(date)
f1.txACT_CHECK_OUT.value=f1.txACT_CHECK_OUT.value & "/" & MONTH(date)
f1.txACT_CHECK_OUT.value=f1.txACT_CHECK_OUT.value & "/" & DAY(date)
end sub
--></script><input
type="submit" name="BTSUBMIT" value="Submit"> <input
type="reset" value="Reset Form"> </p>
<p align="center">&nbsp;</p>
</form>
<br><a href=prot.htm>Back</a><br>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 16, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41187" --></h5>
</body>
</html>

```

Checkout_list.html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=iso-8859-1">

<meta name="FORMATTER" content="Microsoft FrontPage 2.0">

<meta name="GENERATOR" content="Microsoft FrontPage 2.0">

<title>Frontdesk - Return registration list</title>

</head>

<body background="bg080.jpg">

<h1>FRONT DESK CLECK-Return
registration list</h1>

<p>Prepare the room
registration at the end of the day</p>

<hr>

<form action="http://cicc09/hotel/scripts/regis_in.pl"
method="POST" name="f1">

<div align="center"><center><table border="1">

<tr>

<td>DATE</td>

<td><input type="text" size="15" name="TXDATE"></td>

</tr>

</table>

</center></div><p align="center"><input type="submit"
name="BTSUBMIT" value="Submit"><input type="reset"
name="BTRESET" value="Reset"></p>

</form>

<p><script language="VBScript"><!--

sub window_onload

f1.txdate.value=year(date)

f1.txdate.value=f1.txdate.value & "/" & month(date)

f1.txdate.value=f1.txdate.value & "/" & day(date)

end sub

--></script></p>

</body>

</html>

Clerk_registration_report.html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

```

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>FRONT DESK CLERK-Registration report at the end of the day</title>
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="5">FRONT DESK CLERK-Registration
report </font></h1>

<table border="1">
<tr>
<td><strong>REGISTRATION DATE : </strong></td>
<td><font size="2"><strong><input type="text" size="12"
name="TXDATE" value="1997/12/18"></strong></font></td>
</tr>
</table>

<hr>
<div align="center"><center>

<table border="1">
<tr>
<td><font size="2"><strong>ROOM NO.</strong></font></td>
<td><font size="2"><strong>REGISTER ID :</strong></font></td>
</tr>
<tr>
<td><font size="2"><strong><input type="text" size="5"
name="TXROOM_NO" value="101"></strong></font></td>
<td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890101"></strong></font></td>
</tr>
<tr>
<td><font size="2"><strong><input type="text" size="5"
name="TXROOM_NO" value="102"></strong></font></td>
<td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890102"></strong></font></td>
</tr>
<tr>
<td><font size="2"><strong><input type="text" size="5"
name="TXROOM_NO" value="103"></strong></font></td>
<td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890103"></strong></font></td>
</tr>
</table>

```

```

<td><font size="2"><strong><input type="text" size="5"
name="TXROOM_NO" value="104"></strong></font></td>
<td><font size="2"><strong><input type="text" size="5"
name="TXREGISTER_ID" value="1234567890104"></strong></font></td>
</tr>
<tr>
<td><font size="2"><strong><input type="text" size="5"
name="TXROOM_NO" value="105"></strong></font></td>
<td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890105"></strong></font></td>
</tr>
</table>
</center></div>

<hr>

<h5 align="center"><a href="mainmenu.htm">Back to Menu</a></h5>
</body>
</html>

```

Expected_checkout.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>EXPECTED CHECK OUT-Check out tomorrow</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="3">EXPECTED CHECK OUT-Check out
tomorrow<script language="VBScript"><!--
sub window_onload
f1.txdate.value=YEAR(date)
f1.txdate.value=f1.txdate.value & "/" & MONTH(date)
f1.txdate.value=f1.txdate.value & "/" & DAY(date) + 1
end sub
--></script></font></h1>

<form action="/hotel/scripts/exp_chkot.pl"
method="POST" name="F1" onsubmit>
<div align="center"><center><table border="1">
<tr>
<td><strong>CHECK OUT DATE</strong></td>

```

```
 <font size="1"><input type="text" size="10" name="TXDATE"><strong>YYYY/MM/DD</strong> </font></td> </tr> </table> </center></div><p align="center"><input type="submit" name="BTSUBMIT" value="Submit"> <input type="reset" value="Reset Form"> </p> </form>  <br> <a href=prot.htm>Back</a> <hr> <h5>Copyright information goes here.<br> Last revised: <!--webbot bot="TimeStamp" s-type="EDITED" s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot bot="TimeStamp" endspan i-checksum="41171" --></h5> </body> </html> |
```

Extened.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>EXTENDING-Extending the room</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="5">EXTENDING - Extending the room</font></h1>

<hr>

<p><font color="#FF0080" size="5">The guest wants to extend the
stay period.</font>.</p>

<form action="http://cicc09/hotel/scripts/extend.pl"
method="POST" name="f1">
<input type="hidden" name="HDEXTINDATE" value="1"><input
type="hidden" name="HDEXTOUTDATE" value="1"><input
type="hidden" name="HDROOMNO" value="1"><p align="center"> <div
align="center"><center></p>
<table border="1">
<tr>

```

```

        <td><strong>ROOM NO.</strong></td>
        <td colspan="3"><input type="text" size="20"
        name="TXROOMNO"></td>
    </tr>
    <tr>
        <td><strong>EXTEND DATE FROM</strong></td>
        <td><input type="text" size="4" name="TXEXTINYEAR">YYYY</td>
        <td><input type="text" size="2" name="TXEXTINMONTH">MM</td>
        <td><input type="text" size="2" name="TXEXTINDAY">DD</td>
    </tr>
    <tr>
        <td><strong>EXTEND DATE TO </strong></td>
        <td><input type="text" size="4" name="TXEXTOUTYEAR">
        YYYY </td>
        <td><input type="text" size="2" name="TXEXTOUTMONTH">
        MM </td>
        <td><input type="text" size="2" name="TXEXTOUTDAY">
        DD </td>
    </tr>
</table>
</div>
<p align="center"><script language="VBScript"><!--
sub window_onload
F1.txEXTINyear.value=year(date)
F1.txEXTINmonth.value=month(date)
F1.txEXTINday.value=day(date)+1
F1.txROOMNO.focus
F1.txEXTOUTyear.value=year(date)
F1.txEXTOUTmonth.value=month(date)
end sub

sub btsbmit_onclick
f1.hdEXTINdate.value=f1.txEXTINyear.value & "/" & f1.txEXTINmonth.value & "/" & f1.txEXTINday.value
msgbox f1.hdEXTINdate.value
f1.hdEXToutdate.value=f1.txEXToutyear.value & "/" & f1.txEXToutmonth.value & "/" & f1.txEXToutday.value
msgbox f1.hdEXToutdate.value
f1.hdROOMNO.value=F1.TXROOMNO.VALUE
msgbox F1.TXROOMNO.VALUE

end sub

--></script>
</p>
<p align="center"><input type="submit" name="BTSUBMIT"
value="EXTEND"> </p>
</form>

<hr>

<h5>Copyright information goes here.<br>

```

```
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 15, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41183" --></h5>
</body>
</html>
```

Extended_ok.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>EXTENDOK!-Extend the room finish</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="5">EXTENDING OK! - Update done</font></h1>

<hr>

<p><font color="#FF0080" size="5">The room was extending already.</font>.</p>

<form action="http://cicc09/hotel/scripts/extend.pl"
method="POST" name="f1">
  <input type="hidden" name="HDEXTINDATE" value="1"><input
type="hidden" name="HDEXTOUTDATE" value="1"><input
type="hidden" name="HROOMNO" value="1"><div align="center"><center><table
border="1">
  <tr>
    <td><strong>ROOM NO.</strong></td>
    <td colspan="3"><input type="text" size="20"
name="TXROOMNO"></td>
  </tr>
  <tr>
    <td><strong>EXTEND DATE FROM</strong></td>
    <td><input type="text" size="4" name="TXEXTINYEAR">YYYY</td>
    <td><input type="text" size="2" name="TXEXTINMONTH">MM</td>
    <td><input type="text" size="2" name="TXEXTINDAY">DD</td>
  </tr>
  <tr>
    <td><strong>EXTEND DATE TO</strong></td>
    <td><input type="text" size="4" name="TXEXTOUTYEAR">
YYYY</td>
```

```

        <td><input type="text" size="2" name="TXEXTOUTMONTH">
        MM </td>
        <td><input type="text" size="2" name="TXEXTOUTDAY">
        DD </td>
    </tr>
</table>
</center></div><p align="center"><script language="VBScript"><!--
sub window_onload
F1.bxEXTINyear.value=year(date)
F1.bxEXTINmonth.value=month(date)
F1.bxEXTINday.value=day(date)+1
F1.bxROOMNO.focus
F1.bxEXTOUTyear.value=year(date)
F1.bxEXTOUTmonth.value=month(date)
end sub

sub btsbmit_onclick
f1.hdEXTINdate.value=f1.txEXTINyear.value & "/" & f1.bxEXTINmonth.value & "/" & f1.bxEXTINday.value
msgbox f1.hdEXTINdate.value
f1.hdEXToutdate.value=f1.bxEXToutyear.value & "/" & f1.bxEXToutmonth.value & "/" & f1.bxEXToutday.value
msgbox f1.hdEXToutdate.value
f1.hdROOMNO.value=F1.TXROOMNO.VALUE
msgbox F1.TXROOMNO.VALUE

end sub

--></script>
</p>
<p align="center"><input type="submit" name="BTSUBMIT"
value="EXTEND"> </p>
</form>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 15, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41183" --></h5>
</body>
</html>

```

Get_reserve_id.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"

```


<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=iso-8859-1">

<meta name="GENERATOR" content="Microsoft FrontPage 2.0">

<title>LAUNDRY- Laundry charge</title>

<meta name="FORMATTER" content="Microsoft FrontPage 2.0">

</head>

<body background="bg080.jpg">

<h1>LAUNDRY-Laundry charge</h1>

<form action="/hotel/scripts/laundry.pl"

method="POST" name="F1" onsubmit>

<div align="center"><center><table border="1">

<tr>

<td>ROOM NO. </td>

<td><select name="TXROOM_NO" size="1">

<option value="101">101</option>

<option value="102">102</option>

<option value="103">103</option>

<option value="104">104</option>

<option value="105">105</option>

<option value="106">106</option>

<option value="107">107</option>

<option value="108">108</option>

<option value="109">109</option>

<option value="110">110</option>

<option value="111">111</option>

<option value="112">112</option>

<option value="113">113</option>

<option value="114">114</option>

<option value="115">115</option>

<option value="116">116</option>

<option value="117">117</option>

<option value="118">118</option>

<option value="119">119</option>

<option value="120">120</option>

<option value="121">121</option>

<option value="122">122</option>

<option value="123">123</option>

<option value="124">124</option>

<option value="125">125</option>

<option value="126">126</option>

<option value="127">127</option>

<option value="128">128</option>

<option value="129">129</option>
<option value="130">130</option>
<option value="131">131</option>
<option value="132">132</option>
<option value="133">133</option>
<option value="134">134</option>
<option value="135">135</option>
<option value="136">136</option>
<option value="137">137</option>
<option value="138">138</option>
<option value="139">139</option>
<option value="140">140</option>
<option value="141">141</option>
<option value="142">142</option>
<option value="143">143</option>
<option value="144">144</option>
<option value="145">145</option>
<option value="201">201</option>
<option value="202">202</option>
<option value="203">203</option>
<option value="204">204</option>
<option value="205">205</option>
<option value="206">206</option>
<option value="207">207</option>
<option value="208">208</option>
<option value="209">209</option>
<option value="210">210</option>
<option value="211">211</option>
<option value="212">212</option>
<option value="213">213</option>
<option value="214">214</option>
<option value="215">215</option>
<option value="216">216</option>
<option value="217">217</option>
<option value="218">218</option>
<option value="219">219</option>
<option value="220">220</option>
<option value="221">221</option>
<option value="222">222</option>
<option value="223">223</option>
<option value="224">224</option>
<option value="225">225</option>
<option value="226">226</option>
<option value="227">227</option>
<option value="228">228</option>
<option value="229">229</option>
<option value="230">230</option>
<option value="231">231</option>
<option value="232">232</option>
<option value="233">233</option>

```

<option value="234">234</option>
<option value="235">235</option>
<option value="236">236</option>
<option value="237">237</option>
<option value="238">238</option>
<option value="239">239</option>
<option value="240">240</option>
<option value="241">241</option>
<option value="242">242</option>
<option value="243">243</option>
<option value="244">244</option>
<option value="245">245</option>
<option value="246">246</option>
<option value="247">247</option>
<option value="248">248</option>
<option value="249">249</option>
<option value="250">250</option>
<option value="251">251</option>
<option value="252">252</option>
<option value="301">301</option>
<option value="302">302</option>
<option value="303">303</option>
</select></td>
<td><font size="1"><strong>DATE </strong></font></td>
<td><font size="1"><input type="text" size="10"
name="TXLAUNDRY_DATE"><strong>YYYY/MM/DD</strong> </font></td>
</tr>
</table>
</center></div><div align="center"><center><table border="1">
<tr>
<td><script language="VBScript"><!--
sub btsubmit_onclick
f1.tx1.value=trim(f1.tx1.value)
f1.tx2.value=trim(f1.tx2.value)
f1.tx3.value=trim(f1.tx3.value)
f1.tx4.value=trim(f1.tx4.value)
msgbox f1.tx1.value
msgbox f1.tx2.value
msgbox f1.tx3.value
msgbox f1.tx4.value
end sub
sub window_onload
f1.txLAUNDRY_date.value=YEAR(date)
f1.txLAUNDRY_date.value=f1.txLAUNDRY_date.value & "/" & MONTH(date)
f1.txLAUNDRY_date.value=f1.txLAUNDRY_date.value & "/" & DAY(date)
end sub
--></script><font
size="1"><strong>CLOTH TYPE</strong></font></td>
<td><font size="1"><strong>QUANTITY</strong></font></td>
</tr>

```

```

<tr>
  <td><font size="1"><strong>T-SHIRTS</strong></font></td>
  <td><font size="1"><strong><input type="text"
    size="10" name="TX1" value="0"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>COATS</strong></font></td>
  <td><font size="1"><strong><input type="text"
    size="10" name="TX2" value="0"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>TROUSERS</strong></font></td>
  <td><font size="1"><strong><input type="text"
    size="10" name="TX3" value="0"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>SUITS</strong></font></td>
  <td><font size="1"><strong><input type="text"
    size="10" name="TX4" value="0"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>SHIRTS</strong></font></td>
  <td><font size="1"><strong><input type="text"
    size="10" name="TX5" value="0"> </strong></font></td>
</tr>
</table>
</center></div><p align="center"><input type="submit"
name="BTSUBMIT" value="Submit"> <input type="reset"
value="Reset Form"> </p>
</form>

<hr>
<a href=prot.htm>Back</a><br>

```

```

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</body>
</html>

```

Laundry_error.html

```

<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML/EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">

```

```

<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>LAUNDRY- Update ERROR!</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body bgcolor="#FFFFFF">

<h1><font color="#FF0000" size="6">LAUNDRY-Update ERROR!</font></h1>
<div align="center"><center>

<table border="1" background="bg080.jpg">
  <tr>
    <td><font size="1"><strong>ROOM NO. </strong></font></td>
    <td><font size="1"><input type="text" size="10"
name="TXROOM_NO" value="121"></font></td>
    <td><font size="1"><strong>DATE </strong></font></td>
    <td><font size="1"><input type="text" size="11"
name="TXLAUNDRY_DATE" value="1997/12/17"><strong>YYYY/MM/DD</strong>
</font></td>
  </tr>
</table>
</center></div><div align="center"><center>

<table border="1" background="bg080.jpg">
  <tr>
    <td><font size="1"><strong>CLOTH TYPE</strong></font></td>
    <td><font size="1"><strong>QUANTITY</strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>SHIRT</strong></font></td>
    <td><font size="1"><input type="text" size="10"
name="TX1" value="5"> </strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>TROUSERS/BLOUSE</strong></font></td>
    <td><font size="1"><input type="text" size="10"
name="TX2" value="4"> </strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>COATS</strong></font></td>
    <td><font size="1"><input type="text" size="10"
name="TX3" value="0"> </strong></font></td>
  </tr>
  <tr>
    <td><font size="1"><strong>WOOLEN</strong></font></td>
    <td><font size="1"><input type="text" size="10"
name="TX4" value="0"> </strong></font></td>
  </tr>
</table>
</center></div>

```

Back to Menu

</p>

<hr>

<h5>Copyright information goes here.

Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"

s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot

bot="TimeStamp" endspan i-checksum="41171" --></h5>

</body>

</html>

Laundry_ok.html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML/EN">

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=iso-8859-1">

<meta name="GENERATOR" content="Microsoft FrontPage 2.0">

<title>LAUNDRY- Update OK!</title>

<meta name="FORMATTER" content="Microsoft FrontPage 2.0">

</head>

<body bgcolor="#FFFFFF">

<h1>LAUNDRY-Update OK!</h1>

<div align="center"><center>

<table border="1" background="bg080.jpg">

<tr>

<td>ROOM NO. </td>

<td><input type="text" size="10"

name="TXROOM_NO" value="121"></td>

<td>DATE </td>

<td><input type="text" size="11"

name="TXLAUNDRY_DATE" value="1997/12/17">YYYY/MM/DD

</td>

</tr>

</table>

</center></div><div align="center"><center>

<table border="1" background="bg080.jpg">

<tr>

<td>CLOTH TYPE</td>

<td>QUANTITY</td>

</tr>

```

<tr>
  <td><font size="1"><strong>SHIRT</strong></font></td>
  <td><font size="1"><strong><input type="text" size="10"
    name="TX1" value="5"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>TROUSERS/BLOUSE</strong></font></td>
  <td><font size="1"><strong><input type="text" size="10"
    name="TX2" value="4"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>COATS</strong></font></td>
  <td><font size="1"><strong><input type="text" size="10"
    name="TX3" value="0"> </strong></font></td>
</tr>
<tr>
  <td><font size="1"><strong>WOOLEN</strong></font></td>
  <td><font size="1"><strong><input type="text" size="10"
    name="TX4" value="0"> </strong></font></td>
</tr>
</table>
</center></div>

<p align="center"><a href="mainmenu.htm">Back to Menu</a><br>
</p>

<hr>

<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</body>
</html>

```

Meal.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>MEAL-Meal charge</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

```

<h1>MEAL-Meal charge</h1>

<form action="/hotel/scripts/mealc.pl"

method="POST" name="F1" onsubmit>

<input type="hidden" name="HDMEAL_AMOUNT" value="1"><input

type="hidden" name="HDMEAL_DATE" value="1"><input

type="hidden" name="HDROOM_NO" value="1"><p align="center"> </p>

<div align="center"><center><table border="1">

<tr>

<td>ROOM NO. </td>

<td><select name="TXROOM_NO" size="1">

<option value="101">101</option>

<option value="102">102</option>

<option value="103">103</option>

<option value="104">104</option>

<option value="105">105</option>

<option value="106">106</option>

<option value="107">107</option>

<option value="108">108</option>

<option value="109">109</option>

<option value="110">110</option>

<option value="111">111</option>

<option value="112">112</option>

<option value="113">113</option>

<option value="114">114</option>

<option value="115">115</option>

<option value="116">116</option>

<option value="117">117</option>

<option value="118">118</option>

<option value="119">119</option>

<option value="120">120</option>

<option value="121">121</option>

<option value="122">122</option>

<option value="123">123</option>

<option value="124">124</option>

<option value="125">125</option>

<option value="126">126</option>

<option value="127">127</option>

<option value="128">128</option>

<option value="129">129</option>

<option value="130">130</option>

<option value="131">131</option>

<option value="132">132</option>

<option value="133">133</option>

<option value="134">134</option>

<option value="135">135</option>

<option value="136">136</option>

<option value="137">137</option>

<option value="138">138</option>

<option value="139">139</option>
<option value="140">140</option>
<option value="141">141</option>
<option value="142">142</option>
<option value="143">143</option>
<option value="144">144</option>
<option value="145">145</option>
<option value="201">201</option>
<option value="202">202</option>
<option value="203">203</option>
<option value="204">204</option>
<option value="205">205</option>
<option value="206">206</option>
<option value="207">207</option>
<option value="208">208</option>
<option value="209">209</option>
<option value="210">210</option>
<option value="211">211</option>
<option value="212">212</option>
<option value="213">213</option>
<option value="214">214</option>
<option value="215">215</option>
<option value="216">216</option>
<option value="217">217</option>
<option value="218">218</option>
<option value="219">219</option>
<option value="220">220</option>
<option value="221">221</option>
<option value="222">222</option>
<option value="223">223</option>
<option value="224">224</option>
<option value="225">225</option>
<option value="226">226</option>
<option value="227">227</option>
<option value="228">228</option>
<option value="229">229</option>
<option value="230">230</option>
<option value="231">231</option>
<option value="232">232</option>
<option value="233">233</option>
<option value="234">234</option>
<option value="235">235</option>
<option value="236">236</option>
<option value="237">237</option>
<option value="238">238</option>
<option value="239">239</option>
<option value="240">240</option>
<option value="241">241</option>
<option value="242">242</option>
<option value="243">243</option>



```

        <option value="244">244</option>
        <option value="245">245</option>
        <option value="246">246</option>
        <option value="247">247</option>
        <option value="248">248</option>
        <option value="249">249</option>
        <option value="250">250</option>
        <option value="251">251</option>
        <option value="252">252</option>
        <option value="301">301</option>
        <option value="302">302</option>
        <option value="303">303</option>
    </select></td>
    <td><font size="1">DATE </font></td>
    <td><font size="1"><input type="text" size="10"
name="TXMEAL_DATE"> YYYY/MM/DD</font></td>
</tr>
</table>
</center></div><div align="center"><center><table border="1">
<tr>
<td><script language="VBScript"><!--
sub btsubmit_onclick
f1.txmeal_amount.value=trim(f1.txmeal_amount.value)
f1.txmeal_date.value=trim(f1.txmeal_date.value)
' f1.txroom_no.value=trim(f1.txroom_no.value)
' msgbox f1.txroom_no.value
msgbox f1.txmeal_date.value
msgbox f1.txmeal_amount.value

f1.hdmeal_amount.value=f1.txmeal_amount.value
end sub
sub window_onload
f1.txmeal_date.value=date
end sub
--></script>Amount</td>
<td><font size="1"><input type="text" size="10"
name="TXMEAL_AMOUNT"> </font></td>
</tr>
</table>
</center></div><p align="center"><input type="submit"
name="BTSSUBMIT" value="Submit"> <input type="reset"
value="Reset Form"> </p>
</form>

<hr>
<a href=prot.htm>Back</a><br>
<h5>Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot

```

```
bot="TimeStamp" endspace i-checksum="41171" --></h5>
</body>
</html>
```

Registration_list.htm

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>REGISTER-Ask report date </title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080" size="5">Report of Registrations</font></h1>

<hr>

<p>.<script language="VBScript"><!--sub window_onloadF1.txyear.value=year(date)F1.txmonth.value=month(date)F1.txday.value=day(date)
F1.txday.focusend sub--></script> </p>

<form action="/hotel/scripts/reg_list.pl"
method="POST" name="f1">
  <input type="hidden" name="HDDATE" value="1"><div
align="center"><center><table border="0">
  <tr>
    <td><strong>REGISTRATION DATE</strong></td>
    <td><input type="text" size="4" name="TXYEAR"> <strong>YYYY</strong>
  </td>
    <td><input type="text" size="2" name="TXMONTH"> <strong>MM</strong>
  </td>
    <td><input type="text" size="2" name="TXDAY"> <strong>DD</strong>
  </td>
  </tr>
</table>
</center></div><p align="center"><script language="VBScript"><!--
sub btreport_onclick
f1.hddate.value=f1.txyear.value & "/" & f1.txmonth.value & "/" & f1.txday.value
msgbox f1.hddate.value
end sub

sub window_onload
f1.txyear.value=year(date)
f1.txmonth.value=month(date)
```

f1.bday.value=day(date)

end sub

--></script>

</p>

<p align="center"><input type="submit" name="BTREPORT"
value="REPORT"> </p>

</form>

<hr>

<h5>Copyright information goes here.

Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"

s-format="%B %d, %Y" startspan -->March 10, 1998<!--webbot
bot="TimeStamp" endspan i-checksum="41163" --></h5>

Back

</body>

</html>

Registration_infor.html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=iso-8859-1">

<meta name="GENERATOR" content="Microsoft FrontPage 2.0">

<title>Registration-Get customer information</title>

<meta name="FORMATTER" content="Microsoft FrontPage 2.0">

</head>

<body background="bg080.jpg">

<h1>Registration-Get customer information</h1>

<div align="center"><center>

<table border="1">

<tr>

<td>RESERVE ID : </td>

<td><input type="text" size="12"

name="TXRESERVE_ID" value="1234567890"></td>

<td>REGISTER ID : </td>

<td><input type="text" size="15"

name="TXREGISTER_ID" value="1234567890101"> </td>

<td>CHECK IN : <input type="text"

size="15" name="TXDATE" value="1997/12/18"></td>

```

</tr>
</table>
</center></div>

<form method="POST" action="_vti_bin/shtml.dll/regis_info.htm"
webbot-action="--WEBBOT-SELF--">
  <!--webbot bot="SaveResults" u-file="formrslt.htm"
  s-format="HTML/DL" b-label-fields="TRUE" startspan --><input
  type="hidden" name="VTI-GROUP" value="0"><!--webbot
  bot="SaveResults" endspan i-checksum="47042" --><div
  align="center"><center><table border="0">
    <tr>
      <td><font size="1">ROOM NO.</font></td>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1">NAME :</font></td>
      <td><font size="1"><input type="text" size="30"
      name="TXPERSON_NAME" value="MR.XXXXXXXXXXXXXXX"></font></td>
    </tr>
    <tr>
      <td><font size="1"><input type="text" size="10"
      name="TXROOM_NO" value="101"></font></td>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1">PASSPORT NO.</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TXPASSPORT_NO" value="K12312345"></font></td>
    </tr>
    <tr>
      <td><font size="1">TYPE</font></td>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1">ID NO.</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TXID_CARD_NO" value="Z123456789"></font></td>
    </tr>
    <tr>
      <td><font size="1"><input type="text" size="10"
      name="TXROOM_TYPE" value="SINGLE"></font></td>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1">ADDRESS</font></td>
      <td><font size="1"><input type="text" size="40"
      name="TXCONTACT_ADDRESS"
      value="123 YOKOHAMA KENSHU CENTER., YOKOHAMA"></font></td>
    </tr>
    <tr>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1"></font>&nbsp;</td>
      <td><font size="1">PHONE</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TX_CONTACT_PHONE" value="054 1234567"></font></td>
    </tr>
    <tr>

```

```

<td><font size="1"></font>&nbsp;</td>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1">CHECK OUT</font></td>
<td><font size="1"><input type="text" size="20"
name="TXACT_CHECK_OUT" value="1997/12/25"></font></td>
</tr>
<tr>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1"></font>&nbsp;</td>
<td><font size="1">PAYMENT</font></td>
<td><font size="1"><select name="PAYMENT_TYPE"
size="1">
<option selected value="CASH">CASH</option>
<option value="COMPANY">COMPANY</option>
<option value="CREDIT CARD">CREDIT CARD</option>
<option value="TRAVEL CHECK">TRAVEL CHECK</option>
</select></font></td>
</tr>
</table>
</center></div><p align="center"><input type="submit"
value="Submit Form"> <input type="reset" value="Reset Form"> </p>
</form>

<hr>

<h5 align="center">Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->March 10, 1998<!--webbot
bot="TimeStamp" endspan i-checksum="41163" --></h5>
</body>
</html>

```

Registration_report.html

```

<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>REGISTER-Report</title>
</head>

<body background="bg080.jpg" bgcolor="#FFFFFF">

<p><font color="#FF0080" size="4"><strong>REGISTRATION-Report at
the end of the day</strong></font></p>

<p><strong>DATE </strong>: <input type="text" size="20" name="T1"></p>

```

<table border="1">

<tr>

<td>REGISTER ID</td>

<td>ROOM TYPE</td>

<td>ROOM NO.</td>

<td>CUSTOMER NAME</td>

<td>SEX</td>

<td>CHECK IN</td>

<td>CHECK OUT</td>

<td>COMPANY</td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>



[illegible]

```

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>

<p align="center"><font size="1">
<A href=prot.htm>Back</A></font></p>
</body>
</html>

```

Registration_update_ok.html

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Registration-Update OK!</title>
<meta name="FORMATTER" content="Microsoft FrontPage 2.0">
</head>

<body background="bg080.jpg">

<h1><font color="#FF0080">Registration-Update OK!</font></h1>
<div align="center"><center>

<table border="1">
    <tr>
        <td><font size="2"><strong>RESERVE ID : </strong></font></td>
        <td><font size="2"><strong><input type="text" size="12"
name="TXRESERVE_ID" value="1234567890"></strong></font></td>
        <td><font size="2"><strong>REGISTER ID :</strong></font></td>
        <td><font size="2"><strong><input type="text" size="15"
name="TXREGISTER_ID" value="1234567890101"> </strong></font></td>
        <td><font size="2"><strong>CHECK IN :<input type="text"
size="15" name="TXDATE" value="1997/12/18"></strong></font></td>
    </tr>

```

```

</table>
</center></div>

<form action="_vti_bin/shtml.dll/regis_upok.htm" method="POST"
webbot-action="--WEBBOT-SELF--">
  <!--webbot bot="SaveResults" u-file="formrslt.htm"
  s-format="HTML/DL" b-label-fields="TRUE" startspan --><input
  type="hidden" name="VTI-GROUP" value="0"><!--webbot
  bot="SaveResults" endspan i-checksum="47042" --><div
  align="center"><center><table border="1">
    <tr>
      <td><font size="1">ROOM NO.</font></td>
      <td><font size="1">NAME :</font></td>
      <td><font size="1"><input type="text" size="30"
      name="TXPERSON_NAME" value="MR.XXXXXXXXXXXXXX"></font></td>
    </tr>
    <tr>
      <td><font size="1"><input type="text" size="10"
      name="TXROOM_NO" value="101"></font></td>
      <td><font size="1">PASSPORT NO.</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TXPASSPORT_NO" value="K12312345"></font></td>
    </tr>
    <tr>
      <td><font size="1">TYPE.</font></td>
      <td><font size="1">ID NO.</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TXID_CARD_NO" value="Z123456789"></font></td>
    </tr>
    <tr>
      <td><font size="1"><input type="text" size="10"
      name="TXROOM_TYPE" value="SINGLE"></font></td>
      <td><font size="1">ADDRESS</font></td>
      <td><font size="1"><input type="text" size="40"
      name="TXCONTACT_ADDRESS"
      value="123 YOKOHAMA KENSU CENTER., YOKOHAMA"></font></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><font size="1">PHONE</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TX_CONTACT_PHONE" value="054 1234567"></font></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><font size="1">CHECK OUT</font></td>
      <td><font size="1"><input type="text" size="20"
      name="TXACT_CHECK_OUT" value="1997/12/25"></font></td>
    </tr>
    <tr>

```

```

<td>&nbsp;</td>
<td><font size="1">PAYMENT</font></td>
<td><font size="1"><select name="PAYMENT_TYPE"
size="1">
  <option selected value="CASH">CASH</option>
  <option value="COMPANY">COMPANY</option>
  <option value="CREDIT CARD">CREDIT CARD</option>
  <option value="TRAVEL CHECK">TRAVEL CHECK</option>
</select></font></td>
</tr>
</table>
</center></div><p align="center"><a href="mainmenu.htm">Back
to Menu</a></p>
</form>

<hr>

<h5 align="center">Copyright information goes here.<br>
Last revised: <!--webbot bot="TimeStamp" s-type="EDITED"
s-format="%B %d, %Y" startspan -->December 12, 1997<!--webbot
bot="TimeStamp" endspan i-checksum="41171" --></h5>
</body>
</html>

```



Server Code(Perl Script)

Assign room number.PL

```
require "cgi_lib.pl";
```

```
&ReadParse();
```

```
$inreserve_id = $in{HDRESERVE_ID};
```

```
$indate = $in{HDCHECKIN};
```

```
$outdate = "1997/12/25";
```

```
&createOK();
```

```
sub createOK()
```

```
{
```

```
    print "HTTP/1.0 200 OK\n";
```

```
    print "Content-type: text/html\n\n";
```

```
    print "<html> ";
```

```
    print "<title>Assign room number";
```

```
    print "</title>";
```

```
    print "<body background='http://cicc09/hotel/bg080.jpg'>";
```

```
    print "<center>";
```

```
    print "<font size='1'>";
```

```
    print "<table border='1'>";
```

```
    print " <tr>";
```

```
    print "    <td>RESERVATION ID : 1234567890 </td>";
```

```
    print "    <td>CHECK IN DATE : 1997/12/16 </td>";
```

```
    print "    <td>CHECK OUT DATE : ";
```

```
    print "        $outdate ";
```

```
    print "    </td>";
```

```
    print "</table>";
```

```
    print "<table border='1'>";
```

```
    print " <tr>";
```

```
    print "    <td>ROOM TYPE</td>";
```

```
    print "    <td>SINGLE</td>";
```

```
    print "    <td>NO.OF ROOMS</td>";
```

```
    print "    <td> 2</td>";
```

```
    print " </tr>";
```

```
    print " <tr>";
```

```
    print "    <td>ROOM TYPE</td>";
```

```
    print "    <td>TWIN</td>";
```

```
    print "    <td>NO. OF ROOMS</td>";
```

```
    print "    <td>1</td>";
```

```
    print " </tr>";
```

```
    print " <tr>";
```

```

print '      <td>ROOM TYPE</td>';
print '      <td>SUITE</td>';
print '      <td>NO. OF ROOMS</td>';
print '      <td>0</td>';
print '    </tr>';
print '  </table>';
print '  <form action="XX.PL" method="POST">';
print '  <table border="1">';
print '    <tr>';
print '      <td>ROOM NO</td>';
print '      <td>ROOM TYPE</td>';
print '      <td>ROOM STATE</td>';
print '      <td>CHOOSED</td>';
print '    </tr>';
print '    <tr>';
print '      <td>101</td>';
print '      <td>SINGLE</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C1"></td>';
print '    </tr>';
print '    <tr>';
print '      <td>201</td>';
print '      <td>SINGLE</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C6"></td>';
print '    </tr>';
print '    <tr>';
print '      <td>225</td>';
print '      <td>SINGLE</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C7"></td>';
print '    </tr>';
print '    <tr>';
print '      <td>250</td>';
print '      <td>TWIN</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C8"></td>';
print '    </tr>';
print '    <tr>';
print '      <td>223</td>';
print '      <td>SINGLE</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C2"></td>';
print '    </tr>';
print '    <tr>';
print '      <td>131</td>';
print '      <td>TWIN</td>';
print '      <td>EMPTY</td>';
print '      <td><input type="checkbox" name="C3"></td>';
print '    </tr>';

```

```

print '    <tr>';
print '        <td>301</td>';
print '        <td>SUITE</td>';
print '        <td>EMPTY</td> ';
print '        <td><input type="checkbox" name="C4"></td> ';
print '    </tr> ';
print ' <tr> ';
print '     <td>303</td> ';
print '     <td>SUITE</td> ';
print '     <td>EMPTY</td> ';
print '     <td><input type="checkbox" name="C5"></td> ';
print ' </tr> ';
print ' </table> ';
print ' </font>';
print ' <input type="submit" value="Submit"> ';
print ' <input type="reset" value="Reset"> ';
print ' </center> ';
print ' </form> ';
print ' </body> ';
print ' </html> ';
}

```

Assign room number.pl

```

require "cgi_lib.pl";
use hotelc;
use hoteldb;
use rooms;
&ReadParse();
$no_single;
$no_twin;
$no_suite;
@single;
@twin;
@suite;
@sstate;
@tstate;
@sustate;
@arr;
@ROOM_TYPE;
$reserve_id;
$single[0] = undef;
$twin[0] = undef;
$suite[0] = undef;
$no_single = 0;
$no_twin = 0;
$no_suite = 0;
$inreserve_id = $in{HDERESERVE_ID};
$DATE = $in{HDCHECKIN};
# $inreserve_id = "0001*SINGLE*5*TWIN*3*SUITE*1";

```

```
@arr = split(/^$/, $inreserve_id);
#&testparm();
#exit;
$reserve_id = $arr[0];
```

```
#$arr[0] = "0";
#$arr[1] = "SINGLE";
#$arr[2] = 3;
#$arr[3] = 'TWIN';
#$arr[4] = 4;
#$DATE = '12/19/97';
#$inreserve_id = "10002";
```

```
if ($arr[1] =~ /SINGLE/i)
```

```
{
    $no_single = $arr[2];
}
```

```
if ($arr[1] =~ /TWIN/i)
```

```
{
    $no_twin = $arr[2];
}
```

```
if ($arr[1] =~ /SUITE/i)
```

```
{
    $no_suite = $arr[2];
}
```

```
if ($arr[3] =~ /SINGLE/i)
```

```
{
    $no_single = $arr[4];
}
```

```
if ($arr[3] =~ /TWIN/i)
```

```
{
    $no_twin = $arr[4];
}
```

```
if ($arr[3] =~ /SUITE/i)
```

```
{
    $no_suite = $arr[4];
}
```

```
if ($arr[5] =~ /SINGLE/i)
```

```
{
    $no_single = $arr[6];
}
```

```
if ($arr[5] =~ /TWIN/i)
```

```
{
    $no_twin = $arr[6];
}
```

```
if ($arr[5] =~ /SUITE/i)
```

```
{
    $no_suite = $arr[6];
}
```



```

print "arr[0] $arr[1]\n";
if ( ($arr[1] =~ /SINGLE/i) || ($arr[1] =~ /TWIN/i) || ($arr[1] =~ /SUITE/i) )
(
print " SINGLE undef \n";
if ( $arr[1] =~ /SINGLE/i)
{
print " single \n";
$ROOM_TYPE[0] = $hotelc::single;
printf "ROOM TYPE %s \n", $ROOM_TYPE[0];
&rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@single, \@sstate);

for ($k=0; $k<scalar(@ROOM_TYPE); $k++)
{
printf "ROOM TYPE %s \n", $ROOM_TYPE[$k];
printf " single no %s \n", $single[$k];
printf "ROOM STATE %s \n", $sstate[$k];
}
}
if ( $arr[1] =~ /TWIN/i)
{
$ROOM_TYPE[0] = $hotelc::twin;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@twin, \@tstate);
}
if ( $arr[1] =~ /SUITE/i)
{
$ROOM_TYPE[0] = $hotelc::suite;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@suite, \@sustate);
}
}

if ( ($arr[3] =~ /SINGLE/i) || ($arr[3] =~ /TWIN/i) || ($arr[3] =~ /SUITE/i) )
{
$ROOM_TYPE[0] = $hotelc::twin;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@twin, \@tstate);
if ( $arr[3] =~ /SINGLE/i)
{
$ROOM_TYPE[0] = $hotelc::single;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@single, \@sstate);
}
if ( $arr[3] =~ /TWIN/i)
{
$ROOM_TYPE[0] = $hotelc::twin;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@twin, \@tstate);
}
if ( $arr[3] =~ /SUITE/i)
{
$ROOM_TYPE[0] = $hotelc::suite;
rooms::RET_ROOM_STATE_LIST2(\@ROOM_TYPE, $DATE, \@suite, \@sustate);
}
}

```

```

}

}

if ( ($arr[5] =~ /SINGLE/i) || ($arr[5] =~ /TWIN/i) || ($arr[5] =~ /SUITE/i) )
{
    $ROOM_TYPE[0] = $hotelc::suite;
    rooms::RET_ROOM_STATE_LIST2(@ROOM_TYPE,$DATE,@suite,@sustate);
    if ( $arr[5] =~ /SINGLE/i)
    {
        $ROOM_TYPE[0] = $hotelc::single;
        rooms::RET_ROOM_STATE_LIST2(@ROOM_TYPE,$DATE,@single,@sustate);
    }
    if ( $arr[5] =~ /TWIN/i)
    {
        $ROOM_TYPE[0] = $hotelc::twin;
        rooms::RET_ROOM_STATE_LIST2(@ROOM_TYPE,$DATE,@twin,@sustate);
    }
    if ( $arr[5] =~ /SUITE/i)
    {
        $ROOM_TYPE[0] = $hotelc::suite;
        rooms::RET_ROOM_STATE_LIST2(@ROOM_TYPE,$DATE,@suite,@sustate);
    }
}

# printf " room no %s state %s single",$single[0],$sstate[0];
# printf " room type $arr[1]\n";
#if ( ($arr[1] =~ /SINGLE/i) || ($arr[1] =~ /TWIN/i) || ($arr[1] =~ /SUITE/i) )
#{
#for ($i =0;$i < scalar(@single) ;$i++)
#{
# printf " room no %s state %s single\n",$single[$i],$sstate[$i];
#}
#}

#if ( ($arr[3] =~ /SINGLE/i) || ($arr[3] =~ /TWIN/i) || ($arr[3] =~ /SUITE/i) )
#{
# for ($i =0;$i < scalar(@twin) ;$i++)
#{
# printf " room no %s state %s twin\n",$twin[$i],$tstate[$i];
#}
#}

&createOK();
sub testparm
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<html> ";
    print "<BODY>\n";
    for ($i =0;$i < scalar(@arr);$i++)

```

```

{
    print "arr[$i] $arr[$i]\n";
}
print "</BODY></HTML>\n"
}

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<html> ";
    print "<title>Assign room number";
    print "\n";
    print "</title>";
    print "<body background=\"../bg080.jpg\">";
    print "\n";
    print "<center>";
    print "\n";
    print "<font size=\"1\">";
    print "\n";
    print "<table border=\"1\">";
    print " <tr>";
    print "\n";
    print " <td>RESERVATION ID : ";
    printf "%s </td>",$reserve_id;
    print "\n";
    print " <td>CHECK IN DATE : ";
    print $DATE;
    print "</td>";
    print "\n";
    print " </tr>";
    print "</table>";
    print "<table border=\"1\">";
    print " <tr>";
    print " <td>ROOM TYPE</td>";
    print " <td>SINGLE</td>";
    print " <td>NO.OF ROOMS</td>";
    print "<td>";
    print $no_single;
    print "</td>";

    print " </tr>";
    print " <tr>";
    print " <td>ROOM TYPE</td>";
    print " <td>TWIN</td>";
    print " <td>NO. OF ROOMS</td>";
    print "<td>";
    print $no_twin;
    print "</td>";
    print " </tr>";
    print " </tr>";
}

```



```

print '      <td>ROOM TYPE</td>';
print '      <td>SUITE</td>';
print '      <td>NO. OF ROOMS</td>';
print '<td>';
print $no_suite;
print '</td>';

print '    </tr>';
print '</table>';
print "\n";
print '<form action="XX.PL" method="POST" name=form1>';
print "\n";

print '<input type="hidden" name="HDSINGLE" value=';
print $no_single;
print '>';
print '<input type="hidden" name="HDTWIN" value=';
print $no_twin;
print '>';
print '<input type="hidden" name="HDSUITE" value=';
print $no_suite;
print '>';
print '<input type="hidden" name="HDRESERVE_ID" value=';
print $inreserve_id;
print '>';

print '<table border="1">';
print '  <tr>';
print '    <th>ROOM NO</th>';
print '    <th>ROOM TYPE</th>';
print '    <th>ROOM STATE</th>';
print '    <th>CHOOSSED</th>';
if ( $no_single != 0 )
{
  for ($i = 0; $i < scalar(@single); $i++)
  {
    print '      </tr>';
    $sstr = $single[$i];
    $sstr .= ",";
    $sstr .= $sstate[$i];
    $sstr .= ",SINGLE";
    print '<td>';
    print $single[$i];
    print '</td>';
    print ' <td>SINGLE</td>';
    if ( $sstate[$i] == $hotelc::rm_empty )
    {
      print '<td>EMPTY</td>';
    }
    else

```

```

{
    print '<td>ASSIGNED/REGISTERED </td>';
}

print ' <td><input type="checkbox" name="CHK" VALUE=';
print $str;
print '></td>';
print '</tr>';
print "\n";
}
}

if ( $no_twin != 0 )
{
    for ($i = 0; $i < scalar(@twin); $i++)
    {
        print '    </tr>';
        $str = $twin[$i];
        $str .= " ";
        $str .= $state[$i];
        $str .= "TWIN";
        print '<td>';
        print $twin[$i];
        print '</td>';
        print ' <td>TWIN</td>';
        if ( $state[$i] == $hotelc::rm_empty )
        {
            print '<td>EMPTY</td>';
        }
        else
        {
            print '<td>ASSIGNED/REGISTERED </td>';
        }
        print ' <td><input type="checkbox" name="CHK" VALUE=';
        print $str;
        print '></td>';
        print '</tr>';
        print "\n";
    }
}

if ( $no_suite != 0 )
{
    for ($i = 0; $i < scalar(@suite); $i++)
    {
        print '    </tr>';
        $str = $suite[$i];
        $str .= " ";
        $str .= $sustate[$i];
        $str .= "SUITE";
        print '<td>';
        print $suite[$i];
        print '</td>';
    }
}

```

```

print ' <td>SUITE</td>';
if ( $sustate[$i] == $hotelc::rm_empty )
{
    print '<td>EMPTY</td>';
}
else
{
    print '<td>ASSIGNED/REGISTERED </td>';
}
print ' <td><input type="checkbox" name="CHK" VALUE=';
print $str;
print '></td>';
print '</tr>';
print "\n";
}
}

print "\n";
print ' </table> ';
print "\n";
print ' </font>';
print "\n";
print ' <input type="Button" value="Submit" name=BTSUBMIT> ';
print "\n";
print ' <input type="reset" value="Reset"> ';
print "\n";
print ' </center> ';
print "\n";
print ' <script language="VBScript"><!--';
print "\n";
print ' SUB BTSUBMIT_ONCLICK\n';
print ' DIM STR, NO_SINGLE, NO_TWIN, NO_SUITE\n';
print ' NO_SINGLE = 0\n';
print ' NO_TWIN = 0\n';
print ' NO_SUITE = 0\n';

print ' FOR I =0 TO FORM1.ELEMENTS.COUNT - 1\n';
print ' IF FORM1.ELEMENTS(I).NAME = "CHK" THEN';
print "\n";
print ' IF FORM1.ELEMENTS(I).CHECKED THEN\n';
print ' MSGBOX " CHECK BOX " & FORM1.ELEMENTS(I).VALUE';
print "\n";
print ' STR = FORM1.ELEMENTS(I).VALUE\n';
print ' IF ( INSTR(STR,"SINGLE" )) THEN';
print "\n";
print ' NO_SINGLE = NO_SINGLE + 1\n';
print ' END IF\n';
print ' IF ( INSTR(STR,"TWIN" )) THEN';

```

```

print "\n";
print " NO_TWIN = NO_TWIN + 1\n";
print " END IF\n";
print ' IF ( INSTR(STR,"SUITE" ) ) THEN';
print "\n";
print " NO_SUITE = NO_SUITE + 1\n";
print " END IF\n";
print " END IF\n";
print " END IF\n";
print " NEXT\n";
print ' MSGBOX "SINGLE " & FORM1.HDSINGLE.VALUE';
print "\n";
print ' MSGBOX "NO OF SINGLE " & NO_SINGLE';
print "\n";
print ' IF (FORM1.HDSINGLE.VALUE <> "0") THEN';
print "\n";
print " IF ( NO_SINGLE <> cint(FORM1.HDSINGLE.VALUE) ) THEN\n";
print ' MSGBOX "SELECT THE SAME NO OF ROOMS FOR SINGLE";
print "\n";
print " EXIT SUB\n";
print " END IF\n";
print " END IF\n";
print ' IF (FORM1.HDTWIN.VALUE <> "0") THEN';
print "\n";
print " IF ( NO_TWIN <> cint(FORM1.HDTWIN.VALUE) ) THEN\n";
print ' MSGBOX "SELECT THE SAME NO OF ROOMS FOR TWIN";
print "\n";
print " EXIT SUB\n";
print " END IF\n";
print " END IF\n";
print ' IF (FORM1.HDSUITE.VALUE <> "0") THEN';
print "\n";
print " IF ( NO_SUITE <> cint(FORM1.HDSUITE.VALUE) ) THEN\n";
print ' MSGBOX "SELECT THE SAME NO OF ROOMS FOR SUITE";
print "\n";
print " EXIT SUB\n";
print " END IF\n";
print " END IF\n";
print " ' FORM1.SUBMIT\n";
print " END SUB\n";
print ' --></script>';
print "\n";
print '</font></p>';
print '</form> ';
print '</body> ';
print '</html> ';
}

```

```
package bill;
use hoteldb;
my (%data);
```

```
sub new
{
    $data{REGISTRATION_ID}=undef;
    $data{PAYMENT_TYPE}=undef;
    $data{TOTAL_MEAL}=undef;
    $data{TOTAL_LAUNDRY}=undef;
    $data{LAST_MIN_CHARGE}=undef;
    $data{SERVICE_CHARGE}=undef;
    $data{ROOM_NO}=undef;
    $data{ROOM_AMOUNT}=undef;
}
```

```
sub RET_TOT_MEAL_LAUNDRY_AMT
{
    local ($REGISTRATION_ID,*TOTAL_MEAL,*TOTAL_LAUNDRY)=@_;
    &hoteldb::RET_TOT_MEAL_LAUNDRY_AMT($REGISTRATION_ID,$TOTAL_MEAL,$TOTAL_LAUNDRY);
}
```

```
sub CLOSE_BILL
{
    local ($REGISTRATION_ID)=@_;
    &hoteldb::UPDATE_BILL($REGISTRATION_ID);
}
1;
```

Cancel_reservation.pl

```
require "cgi_lib.pl";
use reservation;
&ReadParse();
$RESERVE_ID = $in{TXRESERVE_ID};
$CANCELLED = $in{RESET};
```

```
my (%data);
@WRESERVE_ID;
@WNO_OF_SINGLE;
@WNO_OF_TWIN;
@WNO_OF_SUITE;
```

```
&reservation::GET_EXP_CHECK_IN($RESERVE_ID,%data);
#printf "RESERVE_ID %s\n",$RESERVE_ID;
#printf "data RESERVE_ID %s\n",$data{RESERVE_ID};
@arr0 = split (/ /,$data{EXP_CHECK_IN});
@arr1 = split (/ /,$data{EXP_CHECK_OUT});
$WRESERVE_ID[0]=$RESERVE_ID;
```

```

&res_detail::RET_ROOM_DETAIL(@WRESERVE_ID,@WNO_OF_SINGLE,@WNO_OF_TWIN,@WNO_OF_SUITE);
&reservation::CANCEL_RESERVATION($RESERVE_ID,$CANCELLED,$RES_STATE,@WA_ROOM_NO,@WA_ROOM_TYPE);

#printf "date %s\n", $arr[0];

&createOK();

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print " <title>CANCEL-Cancel reservation OK!</title>";
    print " <h1><font color=#FF0080 size=5>CANCEL-Cancel OK!</font></h1>";
    print " <table border=1>";
    print " <tr>";
    print " <td><font size=1>RESERVATION ID : </font></td>";
    print " <td><font size=1><input type=1 text size=10 name=TXRESERVE_ID value=1>";
    print $data{RESERVE_ID};
    print "></font></td>";
    print " <td><font size=1>CHECK IN : </font></td>";
    print " <td><font size=1><input type=1 text size=12 name=TXEXP_CHECK_IN value=1>";
    print $arr[0];
    print "></font></td>";
    print " <td><font size=1>CHECK OUT : </font></td>";
    print " <td><font size=1><input type=1 text size=12 name=TXEXP_CHECK_OUT value=1>";
    print $arr[1];
    print "></font></td></tr>";
    print " <tr>";
    print " <td><font size=1>SINGLE : </font></td>";
    print " <td><font size=1><input type=1 text size=3 name=TXSINGLE VALUE=1>";
    print $WNO_OF_SINGLE[0];
    print "></font></td>";
    print " <td><font size=1>TWIN :</font></td>";
    print " <td><font size=1><input type=1 text size=3 name=TXTWIN VALUE=1>";
    print $WNO_OF_TWIN[0];
    print "></font></td>";
    print " <td><font size=1>SUITE :</font></td>";
    print " <td><font size=1><input type=1 text size=3 name=TXSUITE VALUE=1>";
    print $WNO_OF_SUITE[0];
    print "></font></td></tr>";
    print "</table>";
    print "<table border=1>";
    print " <tr>";
    print " <td><font size=1>NAME : </font></td>";
    print " <td colspan=3><font size=1><input type=1 text size=40 name=TXCUST_NAME value=1>";
    print $data{CUST_NAME};
    print "></font></td>";
    print " </tr>";
    print " <tr>";

```

```

print ' <td><font size="1">ADDRESS : </font></td>';
print ' <td colspan="3"><font size="1"><input type="text" size="40" name="TXCUST_ADDRESS" value="";
print $data(CUST_ADDR);
print "></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">PHONE :</font></td>';
print ' <td><font size="1"><input type="text" size="10" name="TXCUST_TEL" value="";
print $data(CUST_TEL);
print "></font></td>';
print ' <td><font size="1">FAX:</font></td>';
print ' <td><font size="1"><input type="text" size="10" name="TXCUST_FAX" value="";
print $data(CUST_FAX);
print "></font></td>';
print ' <td><font size="1">SEX : </font></td>';
print ' <td><font size="1"><input type="text" size="10" name="TXCUST_SEX" value="";
print $data(CUST_SEX);
print "></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">CORPORATE :</font></td>';
print ' <td><font size="1"><input type="text" size="20" name="TXCORP_NAME" value="";
print $data(CORP_NAME);
print "></font></td>';
print ' </tr>';
print ' </table>';
print ' <a href="./prot.htm">Back to Menu</a><br>';
print ' </body>';
print ' </html>';
}

```

Check_out_detail.pl

```

require "cgi_lib.pl";
use registration;
use bill;
use payment;
use rooms;

&ReadParse();

$ROOM_NO = $in{TXROOM_NO};
$ACT_CHECK_OUT = $in{TXACT_CHECK_OUT};
#$ROOM_NO = '102';
#$ACT_CHECK_OUT = '1997/03/22';
#$LAST_MIN_CHARGE=100;

$LAST_MIN_CHARGE=$in{LAST_MIN};

```

```

&registration::RET_REG_ID($ROOM_NO,$REGISTRATION_ID,$ROOM_TYPE,$ACT_CHECK_IN,$PAYMENT_TYPE,$ROOM_RATE,$PERSON_
NAME);
@arr0 = split (/,,$ACT_CHECK_IN);
@arr1 = split (/,,$ACT_CHECK_OUT);

&bill::RET_TOT_MEAL_LAUNDRY_AMT($REGISTRATION_ID,$TOTAL_MEAL,$TOTAL_LAUNDRY);
%BILL_INFO=(TOTAL_MEAL=>$TOTAL_MEAL,
TOTAL_LAUNDRY=>$TOTAL_LAUNDRY,
LAST_MIN_CHARGE=>$LAST_MIN_CHARGE,
ROOM_RATE=>$ROOM_RATE,
ACT_CHECK_IN=>$ACT_CHECK_IN,
DATE=>$ACT_CHECK_OUT);
&payment::CAL_TOT_AMT(%BILL_INFO,$PAYMENT_AMOUNT,$ROOM_CHARGE,$SERVICE_CHARGE,$TAX);
&bill::CLOSE_BILL($REGISTRATION_ID);
&rooms::CHANGE_STATE($ROOM_NO);
&registration::CLOSE_REGISTRATION($REGISTRATION_ID);
&payment::CLOSE_PAYMENT($REGISTRATION_ID);

#33333333333333333333
#if ($PAYMENT_TYPE > 0)
#{
#&payment::RET_PAY_TYPE_LIST($PAYMENT_TYPE,@WA_CARD_COMPANY,@WA_DISCOUNT);

# &reserror();
&createOK();

#      <td><select name="TXROOM_NO" size="1">
#          <option value="101">101</option>
#          <option value="102">102</option>
#          <option value="103">103</option>
#          <option value="117">117</option>
#          <option value="118">118</option>
#          <option value="302">302</option>
#          <option value="303">303</option>
#      </select></td>

sub createOK()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print "<HTML><BODY BGCOLOR=WHITE>";
print "<title>CHECK OUT - Check out detail</title>";
print "<h1><font color='FF0080' size='5'>CHECK OUT - Check out detail</h1>";
print "<center>";
print "<table border='1' width='80%' bgcolor =yellow>";
print "    <tr>";
print "        <td><font size =1>REGISTER ID: </td>";
print "        <td><font size =1><input type='text' size='20' name='T1' value='";
print $REGISTRATION_ID;

```

```

print "></td>";
print '<td><font size =1>CHECK IN DATE : </td>';
print '<td><font size =1><input type="text" size="20" name="T2" value="';
print $arr0[0];
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>ROOM NO.</td>';
print ' <td><font size =1><input type="text" size="20" name="T3" value="';
print $ROOM_NO;
print "></td>";
print ' <td><font size =1>CHECK OUT DATE : </td>';
print ' <td><font size =1><input type="text" size="20" name="T4" value="';
print $arr1[0];
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>ROOM TYPE</td>';
print ' <td><font size =1><input type="text" size="20" name="T5" VALUE="';
print $ROOM_TYPE;
print "></td>";
print ' <td><font size =1>PAYMENT TYPE</td>';
print ' <td><font size =1><input type="text" size="20" name="T15" value="';
print $PAYMENT_TYPE;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>NAME :</td>';
print ' <td colspan="3"><font size =1><input type="text" size="45" name="T6" value="';
print $PERSON_NAME;
print "></td>";
print ' </tr>';
print ' </table>';
print ' </center></div><div align="center"><center><table border="1" bgcolor =yellow>';
print ' <tr>';
print ' <td><font size =1>ROOM CHARGE</td>';
print ' <td><font size =1><input type="text" size="20" name="T7" VALUE="';
print $ROOM_CHARGE;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>TOTAL MEAL</td>';
print ' <td><font size =1><input type="text" size="20" name="T8" value="';
print $TOTAL_MEAL;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>TOTAL LAUNDRY</td>';
print ' <td><font size =1><input type="text" size="20" name="T9" value="';
print $TOTAL_LAUNDRY;

```

```

print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>LAST MINUIE CHARGE</td>';
print ' <td><font size =1><input type="text" size="20" name="T10" VALUE="";
print $LAST_MIN_CHARGE;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>SERVICE CHARGE</td>';
print ' <td><font size =1><input type="text" size="20" name="T11" value="";
print $SERVICE_CHARGE;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>TAX</td>';
print ' <td><font size =1><input type="text" size="20" name="T12" VALUE="";
print $TAX;
print "></td>";
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>DISCOUNT</td>';
print ' <td><font size =1><input type="text" size="20" name="T13"></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size =1>TOTAL</td>';
print ' <td><font size =1><input type="text" size="20" name="T14" value="";
print $PAYMENT_AMOUNT;
print "></td>";
print ' </tr>';
print ' </table></font>';
print ' <a href="./prot.htm">Back to Menu</a>';
print ' </center>';
print ' </body>';
print ' </html>';
}

```

Clothes.pm

```

package clothes;

use hoteldb;

my (%data);

sub new
{
    $data{CLOTH_ID} = undef;
    $data{CLOTH_TYPE} = undef;
    $data{CLOTH_PRICE} = undef;
}

sub RET_CLOTH_PRICE

```

```
{
local (*CLOTH_ID,*CLOTH_PRICE)=@_;
&hoteldb::RET_CLOTH_PRICE(@CLOTH_ID,@CLOTH_PRICE);
}
1;
```

Check_out_report.pl

```
use registration;
use hotelc;

require "cgi_lib.pl";
&ReadParse();

$date = $in{TXDATE};

@WA_ROOM_NO;

&registration::RET_ACT_CHECK_OUT_LIST($DATE,@WA_ROOM_NO);

if ( $registration::status==$hotelc::stat_ok )
{ &createOK(); }
else { &reserror(); }

sub createOK()
(
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print "<HTML><BODY BGCOLOR=WHITE>";

print "<title>CHECK OUT REPORT- Check out report</title>";
print "<p><font color=FF0080 size=4>CHECK OUT REPORT- Check out report at the end of the day</font></p>";
print "<p>DATE: <input type='text' size=20 name='T1'></p>";
print "<table border='1'>";
print ' <tr>';
print ' <td>REGISTER ID</td>';
print ' <td>ROOM TYPE</td>';
print ' <td>ROOM NO.</td>';
print ' <td>CUSTOMER NAME</td>';
print ' <td>SEX</td>';
print ' <td>CHECK IN</td>';
print ' <td>CHECK OUT</td>';
print ' <td>COMPANY</td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
```



```

print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' </tr>';
print ' </table>';
print ' <a href=../prot.htm>Back to Menu</a><br>';
print '</body>';
print '</html>';

```



```

}

sub reserror()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<html>';
print '<head>';
print '<meta http-equiv="Content-Type";';
print 'content="text/html; charset=iso-8859-1">';
print '<meta name="GENERATOR" content="Microsoft FrontPage 2.0">';
print '<title>Check-out List</title>';
print '</head>';
print '<body bgcolor="#FFFFFF">';
print '<p><font color="#FF0080" size="5">Check-out List - No list Produced</font></p>';
print '<p>&nbsp;</p>';
print '<hr>';
print '<p align="center"><a href="/hotel/exp_chkot.htm">Back to Reservation</a></p>';
print '</body>';
print '</html>';
}

```

Expect_check_out.pl

```

require "cgi_lib.pl";

use rooms;

&ReadParse();

$DATE = $in(TXDATE);
@WA_ROOMS;
&rooms::RET_CHK_OUT_ROOM_LIST($DATE,@WA_ROOMS);

&createOK();

sub createOK()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<HTML><BODY BGCOLOR=WHITE>';

print '<title>CHECK OUT REPORT- Check out report</title>';
print '<p><font color="#FF0080" size="4">CHECK OUT REPORT- Check out report at the end of the day</font></p>';
print '<p>DATE: <input type="text" size="15" name="T1" value="';
print $DATE;
print '></p>';
print '<table border="1">';

```

```

print ' <tr>';
print ' <td>ROOM NO</td>';
print ' </tr>';

$nn=scalar(@WA_ROOMS);
for($idx=0;$idx<$nn;$idx++)
{
# printf "rooms %s \n", $WA_ROOMS[$idx];
print ' <tr>';
print ' <td>';
print @WA_ROOMS[$idx];
print '</td>';
print ' </tr>';
}

print '</table>';
print ' <a href=../prot.htm>Back to Menu</a><br>';
print '</body>';
print '</html>';
}

```

Extended.pl

```

require "cgi_lib.pl";
&ReadParse();

$room_no = $in{HDROOMNO};
$extindate = $in{HDEXTINDATE};
$extoutdate = $in{HDEXTOUTDATE};

#&extERROR();
&createOK();

sub createOK()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n";
print '<HTML><BODY BGCOLOR=WHITE>';
print '<html>';
print '<title>EXTENDING OK!-Update done</title>';
print '<body background="bg080.jpg">';
print '<h1><font color="#FF0080" size="5">EXTENDING OK! - Extending the room done</h1><hr>';
print '<p>The room was extending already</font></p>';
print '<center><table border="1">';
print ' <tr><td>ROOM NO.</td>';
print ' <td colspan="3"><input type="text" size="20" name="TXROOMNO"></td>';
print ' </tr><tr>';
print ' <td>EXTEND DATE FROM</td>';
print ' <td><input type="text" size="4" name="TXEXTINYEAR">YYYY</td>';
print ' <td><input type="text" size="2" name="TXEXTINMONTH">MM</td>';

```

```

print ' <td><input type="text" size="2" name="TXEXTINDAY">DD</td>';
print ' </tr><tr>';
print ' <td>EXTEND DATE TO </td>';
print ' <td><input type="text" size="4" name="TXEXTOUTYEAR"> YYYY </td>';
print ' <td><input type="text" size="2" name="TXEXTOUTMONTH"> MM </td>';
print ' <td><input type="text" size="2" name="TXEXTOUTDAY"> DD </td>';
print ' </tr></table><a href="mainmenu.htm">Back to Menu</a>';
print '</body></html>';
}

sub extERROR()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<HTML><BODY BGCOLOR=WHITE>';
print '<html>';
print '<title>EXTENDING CANCEL!-Cancel extending</title>';
print '<body background="bg080.jpg">';
print '<h1><font color=red size="5">EXTENDING CANCELED! - Extending the room cancel</h1><hr>';
print '<p>The extending room was cancel</font></p>';
print '<center><table border="1">';
print ' <tr><td>ROOM NO.</td>';
print ' <td colspan="3"><input type="text" size="20" name="TXROOMNO"></td>';
print ' </tr><tr>';
print ' <td>EXTEND DATE FROM</td>';
print ' <td><input type="text" size="4" name="TXEXTINYEAR">YYYY</td>';
print ' <td><input type="text" size="2" name="TXEXTINMONTH">MM</td>';
print ' <td><input type="text" size="2" name="TXEXTINDAY">DD</td>';
print ' </tr><tr>';
print ' <td>EXTEND DATE TO </td>';
print ' <td><input type="text" size="4" name="TXEXTOUTYEAR"> YYYY </td>';
print ' <td><input type="text" size="2" name="TXEXTOUTMONTH"> MM </td>';
print ' <td><input type="text" size="2" name="TXEXTOUTDAY"> DD </td>';
print ' </tr></table><a href="mainmenu.htm">Back to Menu</a>';
print '</body></html>';
}

```

Get_reservation_id.pl

```

require "cgi_lib.pl";
&ReadParse();
$indate = $in(HDDATE);
&createOK;

sub createOK()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<HTML><BODY BGCOLOR=WHITE>';

```

```

print '<title>ASSIGN-Get reserve id by date</title>';
print '<h1>ASSIGN-Assign rooms</h1>';
print '<p>SHOW THE RESERVATION ID FOR CHOOSE ONE</p>';
print '<hr>';
print '<font size = "1">';
print '<center>';
print '<form action="http://cicc09/hotel/scripts/assign_room_no.pl" method="POST">';
print '<table border="1">';
print '    <tr>';
print '        <td><font size="1">RESERVE ID</font></td>';
print '        <td><font size="1">ROOM TYPE</font></td>';
print '        <td><font size="1">NO. OF ROOM</font></td>';
print '        <td><font size="1">ROOM TYPE</font></td>';
print '        <td><font size="1">NO. OF ROOM</font></td>';
print '        <td><font size="1">ROOM TYPE</font></td>';
print '        <td><font size="1">NO. OF ROOM</font></td>';
print '        <td><font size="1">ROOM TYPE</font></td>';
print '        <td><font size="1">NO. OF ROOM</font></td>';
print '    </tr>';
print '</table>';
print '<select name="reserve_id" size="10">';
print '<option select value="999999999 * SINGLE * 999*TWIN * 99 * SUITE * 9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9';
print '</option>';
print '<option value="999999999*SINGLE*999*TWIN*99*SUIE*9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9</option>';
print '<option value="999999999*SINGLE*999*TWIN*99*SUIE*9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9</option>';
print '<option value="999999999*SINGLE*999*TWIN*99*SUIE*9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9</option>';
print '<option value="999999999*SINGLE*999*TWIN*99*SUIE*9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9</option>';
print '<option value="999999999*SINGLE*999*TWIN*99*SUIE*9">999999999 *SINGLE *999 *TWIN *99 *SUITE *9</option>';
print '</select></p>';
print '</font>';
print '<input type="submit" value="Submit Form">';
print '<input type="reset" value="Reset Form"> </p>';
print '</center>';
print '</form>';
print '</html>';
}

```

Get_reserve_id_by_date.pl

```

require "cgi_lib.pl";
use reservation;
use res_detail;
use hotelc;
use hoteldb;
@RESERVE_ID;
&ReadParse();
$indate = ${in{HDDATE}};
# $indate = "1997/8/21";

```

```

&reservation::RET_RESERVE_ID_LIST($indate,@RESERVE_ID);
print "getres $RESERVE_ID[0]\n";
print "getres $RESERVE_ID[1]\n";
if ( $RESERVE_ID[0] == 0 )
{
    &NoAssign();
}
else
{
    &createOK;
}
sub NoAssign
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print "They are no reservations requiring assignment of rooms";
    print "<A HREF=/hotel/assign.htm>Back to Assign Rooms</A>";
    print "</BODY></HTML>";
}
sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print "<title>ASSIGN-Get reserve id by date</title>";
    print "<h1>ASSIGN-Assign rooms</h1>";
    print "<p>SHOW THE RESERVATION ID FOR CHOOSE ONE</p>";
    print "<hr>";
    print "<font size = \"1\">";
    print "<center>";
    print "<form action=\"/hotel/scripts/assroom.pl\" method=\"POST\">";
    printf "<input type=hidden name=HDCHECKIN value=%s >",$indate;
    print "<table border=\"1\">";
    print "    <tr>";
    print "        <td><font size=\"1\">RESERVE ID</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "    </tr>";
    print "</table>";
    print "<select name=\"HDRESERVE_ID\" size=\"10\">";
    print "    ";
    @NO_OF_SINGLE;
    @NO_OF_TWIN;

```

```

&reservation::RET_RESERVE_ID_LIST($indate,@RESERVE_ID);
print " getres $RESERVE_ID[0]\n";
print " getres $RESERVE_ID[1]\n";
if ( $RESERVE_ID[0] == 0 )
{
    &NoAssign();
}
else
{
    &createOK;
}
sub NoAssign
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print "They are no reservations requiring assignment of rooms";
    print "<A HREF=/hotel/assign.htm>Back to Assign Rooms</A>";
    print "</BODY></HTML>";
}
sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print "<title>ASSIGN-Get reserve id by date</title>";
    print "<h1>ASSIGN-Assign rooms</h1>";
    print "<p>SHOW THE RESERVATION ID FOR CHOOSE ONE</p>";
    print "<hr>";
    print "<font size = \"1\">";
    print "<center>";
    print "<form action=\"/hotel/scripts/assroom.pl\" method=\"POST\">";
    printf "<input type=hidden name=HDCHECKIN value=%s >",$indate;
    print "<table border=\"1\">";
    print "    <tr>";
    print "        <td><font size=\"1\">RESERVE ID</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "        <td><font size=\"1\">ROOM TYPE</font></td>";
    print "        <td><font size=\"1\">NO. OF ROOM</font></td>";
    print "    </tr>";
    print "</table>";
    print "<select name=\"HDRESERVE_ID\" size=\"10\">";
    print "\n";
    @NO_OF_SINGLE;
    @NO_OF_TWIN;

```

```

@NO_OF_SUITE;

$display;
$option;

&res_detail::RET_ROOM_DETAIL(@RESERVE_ID,@NO_OF_SINGLE,@NO_OF_TWIN,@NO_OF_SUITE);

# printf "resreve_id $RESERVE_ID[0]\n";
# printf "NO_OF_SINGLE $NO_OF_SINGLE[0]\n";
# printf "ROOM_TWIN $NO_OF_TWIN[0]\n";
# printf "ROOM_suite $NO_OF_SUITE[0]\n";
for ($i =0;$i < scalar(@RESERVE_ID); $i++)
{
    $display = "$RESERVE_ID[$i]**";
    if ( $NO_OF_SINGLE[0] != "0" )
    {
        if ( $NO_OF_SINGLE[$i] != undef )
        (
            $display .= "SINGLE**";
            $display .= $NO_OF_SINGLE[$i];
            $display .= "**";
        )
    }
    if ( $NO_OF_TWIN[0] != "0" )
    {
        if ( $NO_OF_TWIN[$i] != undef )
        {
            $display .= "TWIN**";
            $display .= $NO_OF_TWIN[$i];
            $display .= "**";
        }
    }
    if ( $NO_OF_SUITE[0] != "0" )
    {
        if ( $NO_OF_SUITE[$i] != undef )
        {
            $display .= "SUITE**";
            $display .= $NO_OF_SUITE[$i];
            $display .= "**";
        }
    }
    $option = "<option value=";
    $option .= $display;
    $option .= ">";
    $option .= $display;
    $option .= "</option>\n";
    print $option;
}

print '</select></p>';
print '</font>';
print '<input type="submit" value="Submit Form">';
print '<input type="reset" value="Reset Form"> </p>';

```

```

print '</center>';
print '</form>';
print '</html>';
}

```

Hotelc.pm

```

package hotelc;

*hotelc::single = \0;
*hotelc::twin = \1;
*hotelc::suite = \2;

*hotelc::cash = \0;
*hotelc::tc = \1;
*hotelc::card = \2;
*hotelc::corp = \3;

*hotelc::res_reserved = \0;
*hotelc::res_assigned = \1;
*hotelc::res_closed = \2;
*hotelc::res_cancelled = \3;

*hotelc::rm_assigned = \0;
*hotelc::rm_registered = \1;
*hotelc::rm_empty = \2;
*hotelc::rm_asg_checkin = \3;

*hotelc::reg_closed = \0;
*hotelc::reg_open = \1;

*hotelc::pay_closed = \0;
*hotelc::pay_open = \1;

*hotelc::bill_closed = \0;
*hotelc::bill_open = \1;

*hotelc::stat_ok = \0;
*hotelc::stat_err = \1;

1;

```

Hoteldb.pm

```

package hoteldb;
use Win32::ODBC;
use hotelc;
my ($Data);
$hoteldb::status;

```

```

$hoteldb::err_str;

sub _opendb
{
# $Data = new Win32::ODBC("DSN=ntoracle;UID=W10;PWD=W10;");
$Data = new Win32::ODBC("HOTEL");
if ( $Data )
{
return ($hoteldb::stat_ok);
}
else
{
$hoteldb::err_str="DB Open error\n";
return($hoteldb::stat_err);
}
}
#end sub _opendb

sub _get_db_data
{
local($sql_str)=@_;
# printf "SQL string : %s\n",$sql_str;
$rc = $Data->Sql($sql_str);

if ( $rc )
{
$hoteldb::err_str = $error[0];
$hoteldb::err_str .= $error[1];
$hoteldb::err_str .= $error[2];
$Data->Close();
$hoteldb::status=$hoteldb::stat_err;
}
}
#end _get_db_data

sub _db_close
{
$Data->Close();
}

sub RET_DISCOUNT
{
local ($CORP_NAME,*CORP_DISCOUNT)=@_;

$hoteldb::status = _opendb();
if ($hoteldb::status == $hoteldb::stat_ok)
{
$sql_str="SELECT CORP_DISCOUNT FROM CORPORATE";
$sql_str.=" WHERE CORP_NAME=";
$sql_str.=$CORP_NAME;
$sql_str.="";

```

```

_get_db_data($sql_str);
while ($Data->FetchRow())
{
    $CORP_DISCOUNT=$Data->Data();
}

_db_close;
}
}

sub RET_INIT_RES_ID
{
    local (*RESERVE_ID)=@_;
    $hotelb::status = _opendb();
    if ($hotelb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT RESERVE_ID FROM RESKEY";

        _get_db_data($sql_str);
        while ($Data->FetchRow())
        {
            $RESERVE_ID=$Data->Data();
        }

        _db_close;
    }
}

sub UPDATE_RESERVATION
{
    local (*data)=@_;
    $hotelb::status = _opendb();
    if ($hotelb::status == $hotelc::stat_ok)
    {
        $sql_str="INSERT INTO RESERVATION VALUES(";
        $sql_str=$data{RESERVE_ID};
        $sql_str=",";
        $sql_str=$data{CORP_NAME};
        $sql_str=",";
        $sql_str=$data{EXP_CHECK_IN};
        $sql_str=",";
        $sql_str=$data{EXP_CHECK_OUT};
        $sql_str=",";
        $sql_str=$data{CUST_NAME};
        $sql_str=",";
        $sql_str=$data{CUST_ADDR};
        $sql_str=",";
        $sql_str=$data{CUST_SEX};
        $sql_str=",";
        $sql_str=$data{CUST_TEL};
        $sql_str=",";
    }
}

```

```

$sql_str=$data(CUST_FAX);
$sql_str=",";
$sql_str=$data(RES_STATE);
$sql_str=")";

&_get_db_data($sql_str);

&_db_close;
&UPDATE_RESKEY;
}
}
sub UPDATE_RESKEY
{

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    $sql_str="update reskey set reserve_id=reserve_id + 1";
    &_get_db_data($sql_str);

    &_db_close;
}
}
sub RET_RESERVE_ID
{
    local ($DATE,*RESERVE_ID)=@_;
    $flag = 0;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT RESERVATION.RESERVE_ID FROM RESERVATION";
        $sql_str.=" WHERE ((RESERVATION.EXP_CHECK_IN = #";
        $sql_str=$DATE;
        $sql_str.=") AND (RES_STATE =";
        $sql_str.= $hotelc::res_reserved;
        $sql_str.="))";
        # print "hoteldb DATE $DATE\n";
        &_get_db_data($sql_str);

        $i=0;
        while ($Data->FetchRow())
        {
            @arr = $Data->Data();
            $RESERVE_ID[$i] = $arr[0];
            # print "hoteldb ret_reserve_id $RESERVE_ID[$i]\n";
            $flag = 1;
            $i++;
        }

        &_db_close;
    }
}

```

```

    }
    if ( $flag != 1 )
    {
        $RESERVE_ID[0] = 0;
    }
}

sub SET_RES_STATE_ASSIGNED
{
    local ($RESERVE_ID)=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="update RESERVATION set RES_STATE=";
        $sql_str=$hotelc::res_reserved;
        $sql_str.=" where RESERVE_ID=";
        $sql_str=$RESERVE_ID;
        $sql_str="";
        &_get_db_data($sql_str);
        &_db_close;
    }
}

sub GET_EXP_CHECK_IN
{
    local ($RESERVE_ID,*EXP_CHECK_IN)=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT RESERVATION.EXP_CHECK_IN FROM RESERVATION";
        $sql_str.=" WHERE (((RESERVATION.RESERVE_ID)=";
        $sql_str=$RESERVE_ID;
        $sql_str.="))";
        &_get_db_data($sql_str);
        while ($Data->FetchRow())
        {
            $EXP_CHECK_IN =$Data->Data();
        }

        &_db_close;
    }
}

sub SET_RES_STATE_CANCELLED
{
    local ($RESERVE_ID,*RES_STATE)=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="select RES_STATE from RESERVATION";
        $sql_str.=" where (((RESERVE_ID)=";

```

```

$sql_str=$RESERVE_ID;
$sql_str.=")");
&_get_db_data($sql_str);

while ($Data->FetchRow())
{
    $RES_STATE =$Data->Data();
}
}

$sql_str="delete from RESERVATION";
$sql_str.=" where RESERVE_ID=";
$sql_str=$RESERVE_ID;
$sql_str.="";
&_get_db_data($sql_str);

$sql_str="delete from RESERVE_DETAIL";
$sql_str.=" where RESERVE_ID=";
$sql_str=$RESERVE_ID;
$sql_str.="";

&_get_db_data($sql_str);

&_db_close;
}

sub UPDATE_RESERVE_DETAIL
{
    local ($RESERVE_ID,*ROOM_TYPE,*NO_OF_ROOMS)=@_;
    $hoteldb::status = _opendb();

    $length=scalar(@ROOM_TYPE);
    if ($hoteldb::status == $hoteldb::stat_ok)
    {
        for($i=0;$i<$length;$i++)
        (
            $sql_str="INSERT INTO RESERVE_DETAIL VALUES(";
            $sql_str=$RESERVE_ID;
            $sql_str=",";
            $sql_str=$ROOM_TYPE[$i];
            $sql_str=",";
            $sql_str=$NO_OF_ROOMS[$i];
            $sql_str=")";

            &_get_db_data($sql_str);
        )
        &_db_close;
    }
}

sub RET_CLOTH_PRICE

```

```

{
local (*CLOTH_ID,*CLOTH_PRICE)=@_;
$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
$sql_str="SELECT CLOTH_ID,CLOTH_PRICE FROM CLOTHES";

&_get_db_data($sql_str);

$i=0;
while ($Data->FetchRow())
{
    $CLOTH_ID[$i] = $Data->Data(CLOTH_ID);
    $CLOTH_PRICE[$i] = $Data->Data(CLOTH_PRICE);
    $i++;
}

&_db_close;
}
}
sub RET_ROOM_DETAIL
{
local (*RESERVE_ID,*NO_OF_SINGLE,*NO_OF_TWIN,*NO_OF_SUITE)=@_;

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
# print " hoteldb reservid $RESERVE_ID[0]\n";
$NO_OF_SINGLE[0]="0";
$NO_OF_TWIN[0]="0";
$NO_OF_SUITE[0]="0";
for ($i=0;$i<scalar(@RESERVE_ID);$i++)
{
    $sql_str="SELECT NO_OF_ROOMS,ROOM_TYPE FROM RESERVE_DETAIL";
    $sql_str.=" where (((RESERVE_ID)=$i";
    $sql_str.="$RESERVE_ID[$i];
    $sql_str.=")";

    &_get_db_data($sql_str);

    $NO_OF_SINGLE[$i] = "0";
    $NO_OF_TWIN[$i] = "0";
    $NO_OF_SUITE[$i] = "0";
    while ($Data->FetchRow())
    {

        @arr = $Data->Data();
        if ( $arr[1] == $hotelc::single )
        {
            $NO_OF_SINGLE[$i] = $arr[0];

```

```

    }
    if ( $arr[1] == $hotelc::twin )
    {
        $NO_OF_TWIN[$i] = $arr[0];
    }
    if ( $arr[1] == $hotelc::suite )
    {
        $NO_OF_SUITE[$i] = $arr[0];
    }
    # printf "hoteldb NO_OF_ROOMS %s\n",$NO_OF_ROOMS[$k];
    # printf "hoteldb ROOM_TYPE %s\n",$ROOM_TYPE[$k];
    }

}

&_db_close;
}

}

#the following for reservation checking
sub CHECK_AVAIL_ROOMS
{
    local ($EXP_CHECK_IN,$EXP_CHECK_OUT,*ROOM_TYPE,*NO_OF_ROOMS,*WA_AVAIL_NO_OF_ROOMS,*WA_ROOM_RATE)=@_;
    $hoteldb::status = _opendb();
    $length=scalar(@ROOM_TYPE);

    $rm_total[0]=60;
    $rm_total[1]=37;
    $rm_total[2]=3;
    $total;

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        for ($i=0;$i<$length;$i++)
        {
            if ($ROOM_TYPE[$i]==$hotelc::single) {$total=$rm_total[0]}
            if ($ROOM_TYPE[$i]==$hotelc::twin) {$total=$rm_total[1]}
            if ($ROOM_TYPE[$i]==$hotelc::suite) {$total=$rm_total[2]}

            $sql_str="SELECT Sum(NO_OF_ROOMS) ";
            $sql_str.="FROM RESERVE_DETAIL ";
            $sql_str.="WHERE (((RESERVE_DETAIL.RESERVE_ID)";
            $sql_str.="*n (SELECT RESERVATION.RESERVE_ID FROM RESERVATION ";
            $sql_str.="WHERE (((RESERVATION.EXP_CHECK_IN)<=#";
            $sql_str.="EXP_CHECK_IN;
            $sql_str.=") AND ((RESERVATION.EXP_CHECK_OUT)>=#";
            $sql_str.="EXP_CHECK_IN;
            $sql_str.=") OR (((RESERVATION.EXP_CHECK_IN)<=#";
            $sql_str.="EXP_CHECK_OUT;
            $sql_str.=") AND ((RESERVATION.EXP_CHECK_OUT)>=#";

```

```

$sql_str.=$EXP_CHECK_OUT;
$sql_str.="#)))) AND ((RESERVE_DETAIL.ROOM_TYPE)=*";
$sql_str.=$ROOM_TYPE[$i];
$sql_str.=")");

&_get_db_data($sql_str);
$Data->FetchRow();
$WA_AVAIL_NO_OF_ROOMS[$i]=$Data->Data();
$WA_AVAIL_NO_OF_ROOMS[$i]=$total - $WA_AVAIL_NO_OF_ROOMS[$i];
}
}

```

##the following is for access (for this project)

```

# SELECT Sum(no_of_rooms) AS Expr1
# FROM reserve_detail
# WHERE (((reserve_detail.RESERVE_ID)
# In (SELECT RESERVATION.RESERVE_ID
# FROM RESERVATION
# WHERE (((RESERVATION.EXP_CHECK_IN)<=[win])
# AND ((RESERVATION.EXP_CHECK_OUT]>=[win]))
# OR (((RESERVATION.EXP_CHECK_IN)<=[wout])
# - AND ((RESERVATION.EXP_CHECK_OUT]>=[wout])))))
# AND ((reserve_detail.ROOM_TYPE)=[wtype]));

```

following is a sample for oracle - but querying is not for this project

```

# SELECT Sum(NO_OF_ROOMS)
# FROM RESERVATION , RESERVE_DETAIL
# ON RESERVATION.RESERVE_ID = RESERVE_DETAIL.RESERVE_ID
# WHERE ( ((RESERVATION.EXP_CHECK_IN)<=[WEXP_CHECK_IN]) AND
#      (([WEXP_CHECK_IN]<[reservation].[exp_check_out]) AND
#      ((RESERVE_DETAIL.ROOM_TYPE)=[wtype])
#      ) OR
# ( ((RESERVATION.EXP_CHECK_IN)<=[WEXP_CHECK_OUT]) AND
#      (([WEXP_CHECK_OUT]<[reservation].[exp_check_out]) AND
#      RESERVATION.RES_STATE <> 2;
# );

```

```

if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
for ($i=0;$i<$length;$i++)
(
$sql_str="SELECT ROOM_RATE.ROOM_RATE FROM ROOM_RATE WHERE ";
$sql_str.=$ROOM_TYPE[$i];
$sql_str.=$ROOM_TYPE[$i];

&_get_db_data($sql_str);
$Data->FetchRow();
$WA_ROOM_RATE[$i]=$Data->Data();
}
}

```

```
&_db_close;
}
#end sub CHECK_AVAIL_ROOMS
```

```
#the following used in test1.pl
#tested
sub RET_ACT_CHK_OUT_LIST
{
  local ($DATE,*WA_ROOM_NO)=@_;
```

```
  $hoteldb::status = _opendb();
```

```
  if ($hoteldb::status == $hotelc::stat_ok)
```

```
  {
    $sql_str="SELECT REGISTRATION.ROOM_NO FROM REGISTRATION";
    $sql_str.=" WHERE (((REGISTRATION.ACT_CHECK_OUT)=#";
    $sql_str.=$DATE;
    $sql_str.="#))";
```

```
    &_get_db_data($sql_str);
```

```
    $idx=0;
```

```
    while ($Data->FetchRow())
```

```
    {
      $WA_ROOM_NO[$idx] = $Data->Data();
      $idx++;
    }
```

```
    &_db_close;
```

```
  }
}
#end sub RET_ACT_CHK_OUT_LIST
```

```
#the following used in test2.pl
```

```
#tested
```

```
sub CREATE_REGISTRATION
```

```
{
  local($RESERVE_ID,*WA_ROOM_NO,*WA_ROOM_STATE)=@_;
```

```
  $length=scalar(@WA_ROOM_NO);
```

```
  if ($hoteldb::status == $hotelc::stat_ok)
```

```
  {
    for ($i=0;$i<$length;$i++)
```

```

{
    $sql_str="INSERT INTO REGISTRATION ";
    $sql_str.="(RESERVE_ID,ROOM_NO,REG_CLOSED) VALUES (";
    $sql_str.=$RESERVE_ID;
    $sql_str.=",";
    $sql_str.=$WA_ROOM_NO[$i];
    $sql_str.=",";
    $sql_str.=$hotelc::reg_open;
    $sql_str.=");";

```

```

    &_get_db_data($sql_str);

```

```

}
}
if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    for ($i=0;$i<$length;$i++)
    {
        $sql_str="UPDATE ROOMS SET ROOMS.ROOM_STATE=";
        $sql_str.=$WA_ROOM_STATE[$i];
        $sql_str.=" WHERE (ROOMS.ROOM_NO=";
        $sql_str.=$WA_ROOM_NO[$i];
        $sql_str.=");";
        &_get_db_data($sql_str);
    }
}
&_db_close;
}
#end CREATE_REGISTRATION

```

```

#the following used in test3.pl

```

```

sub RET_RESERVED_ROOMS

```

```

#tested

```

```

{
    local ($RESERVE_ID,$WA_ROOM_NO,$WA_ROOM_TYPE)=@_;

```

```

    $sql_str="SELECT REGISTRATION.ROOM_NO ";
    $sql_str.=" FROM REGISTRATION";
    $sql_str.=" WHERE (REGISTRATION.RESERVE_ID=";
    $sql_str.=$RESERVE_ID;
    $sql_str.=");";

```

```

    $hoteldb::status = _opendb();

```

```

    if ($hoteldb::status == $hotelc::stat_ok)
    {

```

```

        &_get_db_data($sql_str);
    }
}

```

```

if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    $idx=0;
    while ($Data->FetchRow())
    {
        $WA_ROOM_NO[$idx] = $Data->Data();
        $idx++;
    }
}

$length=scalar(@WA_ROOM_NO);

for ($i=0;$i<$length;$i++)
{
    $sql_str="SELECT ROOMS.ROOM_TYPE ";
    $sql_str.=" FROM ROOMS";
    $sql_str.=" WHERE (ROOMS.ROOM_NO=";
    $sql_str.=$WA_ROOM_NO[$i];
    $sql_str.=")";
    &_get_db_data($sql_str);
    $Data->FetchRow();
    $WA_ROOM_TYPE[$i] = $Data->Data();
}
&_db_close;
}
#end RET_RESERVED_ROOMS

#the following used in test4.pl
sub UPDATE_REGISTRATION
#tested
{
    local($RESERVE_ID,*CUST_INFO,$ROOM_NO,$PAYMENT_TYPE,*REGISTRATION_ID)=@_;

    $REGISTRATION_ID=$RESERVE_ID;
    $REGISTRATION_ID.=$ROOM_NO;

    $hoteldb::status = _opendb();

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="UPDATE REGISTRATION ";
        $sql_str.=" SET REGISTRATION.REGISTRATION_ID=";
        $sql_str.=$REGISTRATION_ID;
        $sql_str.=",";
        $sql_str.="REGISTRATION.PERSON_NAME=";
        $sql_str.=$CUST_INFO{PERSON_NAME};
        $sql_str.=",";
        $sql_str.="REGISTRATION.PASSPORT_NO=";
        $sql_str.=$CUST_INFO{PASSPORT_NO};
    }
}

```

```

$sql_str.=",";
$sql_str.="REGISTRATION.ID_CARD_NO=";
$sql_str.=$CUST_INFO(ID_CARD_NO);
$sql_str.=",";
$sql_str.="REGISTRATION.CONTACT_ADDR=";
$sql_str.=$CUST_INFO(CONTACT_ADDRESS);
$sql_str.=",";
$sql_str.="REGISTRATION.CONTACT_PHONE=";
$sql_str.=$CUST_INFO(CONTACT_PHONE);
$sql_str.=",";
$sql_str.="REGISTRATION.ACT_CHECK_IN=#";
$sql_str.=$CUST_INFO(ACT_CHECK_IN);
$sql_str.="#,";
$sql_str.="REGISTRATION.ACT_CHECK_OUT=#";
$sql_str.=$CUST_INFO(ACT_CHECK_OUT);
$sql_str.="#,";
$sql_str.="REGISTRATION.PAYMENT_TYPE=";
$sql_str.=$PAYMENT_TYPE;
$sql_str.=" WHERE (REGISTRATION.ROOM_NO=";
$sql_str.=$ROOM_NO;
$sql_str.=" AND ";
$sql_str.="REGISTRATION.RESERVE_ID=";
$sql_str.=$RESERVE_ID;
$sql_str.=")";

&_get_db_data($sql_str);
}

if ($hoteldb::status == $hotelc::stat_ok)
{
    $sql_str="UPDATE ROOMS ";
    $sql_str.=" SET ROOMS.ROOM_STATE=";
    $sql_str.=$hotelc::rm_registered;
    $sql_str.=" WHERE ROOMS.ROOM_NO=";
    $sql_str.=$ROOM_NO;
    $sql_str.="";

    &_get_db_data($sql_str);
}

if ($hoteldb::status == $hotelc::stat_ok)
{
    $sql_str="INSERT INTO BILL ";
    $sql_str.="(REGISTRATION_ID,ROOM_NO,BILL_CLOSED) VALUES (";
    $sql_str.=$REGISTRATION_ID;
    $sql_str.=",";
    $sql_str.=$ROOM_NO;
    $sql_str.=",";
    $sql_str.=$hotelc::bill_open;

```

```

$sql_str.="*";

    &_get_db_data($sql_str);
}

    &_db_close;
}

#end sub UPDATE_REGISTRATION

#the following used in test5.pl
sub RET_REG_LIST
#tested
{
    local ($DATE,*WA_ROOM_NO,*WA_REGISTRATION_ID)=@_;

    $sql_str="SELECT REGISTRATION.ROOM_NO,REGISTRATION.REGISTRATION_ID ";
    $sql_str.=" FROM REGISTRATION";
    $sql_str.=" WHERE (REGISTRATION.ACT_CHECK_IN=#";
    $sql_str.=$DATE;
    $sql_str.="#)";

    $hoteldb::status = _opendb();

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        &_get_db_data($sql_str);
    }

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $idx=0;
        while ($Data->FetchRow())
        {
            $WA_ROOM_NO[$idx] = $Data->Data("ROOM_NO");
            $WA_REGISTRATION_ID[$idx] = $Data->Data("REGISTRATION_ID");
            $idx++;
        }
    }

    &_db_close;
}

#end RET_REG_LIST

#the following used in test6.pl
sub RET_REG_ID
#tested
{
    local ($ROOM_NO,*REGISTRATION_ID,*ROOM_TYPE,*ACT_CHECK_IN,*PAYMENT_TYPE,*ROOM_RATE,*PERSON_NAME)=@_;

```

```

$sql_str="SELECT REGISTRATION.REGISTRATION_ID,;
$sql_str="REGISTRATION.ACT_CHECK_IN,;
$sql_str="REGISTRATION.PAYMENT_TYPE,;
$sql_str="REGISTRATION.PERSON_NAME;
$sql_str=" FROM REGISTRATION";
$sql_str.=" WHERE (REGISTRATION.ROOM_NO=";
$sql_str=$ROOM_NO;
$sql_str.=" AND REGISTRATION.REG_CLOSED=";
$sql_str=$hotelc::reg_open;
$sql_str=")";

```

```

$hoteldb::status = _opendb();

```

```

if ($hoteldb::status == $hotelc::stat_ok)
{
    &_get_db_data($sql_str);
}

```

```

if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    $Data->FetchRow();

    $REGISTRATION_ID = $Data->Data("REGISTRATION_ID");
    $ACT_CHECK_IN = $Data->Data("ACT_CHECK_IN");
    $PAYMENT_TYPE = $Data->Data("PAYMENT_TYPE");
    $PERSON_NAME = $Data->Data("PERSON_NAME");
}
@dt=split(//,$ACT_CHECK_IN);
$ACT_CHECK_IN=$dt[0];

```

```

$sql_str="SELECT ROOMS.ROOM_TYPE,;
$sql_str="ROOMS.ROOM_RATE;
$sql_str=" FROM ROOMS";
$sql_str.=" WHERE (ROOMS.ROOM_NO=";
$sql_str=$ROOM_NO;
$sql_str=")";

```

```

if ($hoteldb::status == $hotelc::stat_ok)
{
    &_get_db_data($sql_str);
}

```

```

if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    $Data->FetchRow();
    $ROOM_TYPE = $Data->Data("ROOM_TYPE");
    $ROOM_RATE = $Data->Data("ROOM_RATE");
}

```

```

&_db_close;

```

```

}

#end sub RET_REG_ID

##the following used in test7.pl
sub CLOSE_REGISTRATION
#tested
{
    local($REGISTRATION_ID)=@_;

    $sql_str="UPDATE REGISTRATION ";
    $sql_str.=" SET REGISTRATION.REG_CLOSED=";
    $sql_str.=$hotelc::reg_closed;
    $sql_str.=" WHERE REGISTRATION.REGISTRATION_ID=";
    $sql_str.=$REGISTRATION_ID;
    $sql_str.="";

    $hoteldb::status = _opendb();

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        &_get_db_data($sql_str);
    }
    &_db_close;
}
#end sub CLOSE_REGISTRATION

sub CANCEL_REGISTRATION
#tested
{
    local ($RESERVE_ID)=@_;
    $sql_str="DELETE FROM REGISTRATION ";
    $sql_str.="WHERE REGISTRATION.RESERVE_ID=";
    $sql_str.=$RESERVE_ID;
    $sql_str.="";

    $hoteldb::status = _opendb();

    if ($hoteldb::status == $hotelc::stat_ok)
    {
        &_get_db_data($sql_str);
    }
    &_db_close;
}
#----- ORKUN'S SERVICES-----
sub RET_TOT_MEAL_LAUNDRY_AMT
{
    local ($REGISTRATION_ID,*TOTAL_MEAL,*TOTAL_LAUNDRY)=@_;

```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    $sql_str="select BILL.TOTAL_MEAL,BILL.TOTAL_LAUNDRY from BILL";
    $sql_str.=" where (REGISTRATION_ID=\\";
    $sql_str=$REGISTRATION_ID;
    $sql_str.="\\)";
    &_get_db_data($sql_str);
    $status = $Data->FetchRow();
    @arr = $Data->Data();
    $TOTAL_MEAL = $arr[0];
    $TOTAL_LAUNDRY = $arr[1];
    &_db_close;
}
}
sub UPDATE_ACT_CHK_OUT
(
    local ($REGISTRATION_ID) = @_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    (
        $sql_str="UPDATE REGISTRATION SET REGISTRATION.";
        $sql_str.="ACT_CHECK_OUT = Date()";
        $sql_str.=" where (REGISTRATION_ID\\";
        $sql_str=$REGISTRATION_ID;
        $sql_str.="\\)";
        &_get_db_data($sql_str);
        &_db_close();
    )
}
sub UPDATE_BILL
{
    local ($REGISTRATION_ID)=@_;

    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="UPDATE BILL SET BILL.BILL_CLOSED = 0";
        $sql_str.=" where (REGISTRATION_ID=\\";
        $sql_str=$REGISTRATION_ID;
        $sql_str.="\\)";
        &_get_db_data($sql_str);
        &_db_close;
        &UPDATE_ACT_CHK_OUT($REGISTRATION_ID);
    }
}

sub UPT_MEAL_AMT
{
    local (*data) = @_;

```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    $REGISTRATION_ID = $data{REGISTRATION_ID};
    $MEAL_AMOUNT = $data{MEAL_AMOUNT};
    $sql_str="insert into MEAL VALUES(";
    $sql_str=$data{REGISTRATION_ID};
    $sql_str="",#";
    $sql_str=$data{MEAL_DATE};
    $sql_str="",#";
    $sql_str=$data{MEAL_AMOUNT};
    $sql_str="",#";
    $sql_str=$data{MEAL_DESC};
    $sql_str="",#";
    $sql_str=$data{ROOM_NO};
    $sql_str=")";
    &_get_db_data($sql_str);
    &_db_close;
    &ADD_MEAL_AMT{$REGISTRATION_ID,$MEAL_AMOUNT};
}
}
sub UPT_LAUND_AMT
{
    local (*data) = @_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $REGISTRATION_ID = $data{REGISTRATION_ID};
        $LAUND_AMOUNT = $data{LAUND_AMOUNT};
        $sql_str="insert into LAUNDRY VALUES(";
        $sql_str=$data{REGISTRATION_ID};
        $sql_str="",#";
        $sql_str=$data{LAUND_DATE};
        $sql_str="",#";
        $sql_str=$data{LAUND_AMOUNT};
        $sql_str="",001',1,";
        $sql_str=$data{ROOM_NO};
        $sql_str=")";
        &_get_db_data($sql_str);
        &_db_close;
        &ADD_LAUND_AMT{$REGISTRATION_ID,$LAUND_AMOUNT};
    }
}

sub ADD_MEAL_AMT
{
    local ($REGISTRATION_ID,$MEAL_AMOUNT)=@_;

    $hoteldb::status = _opendb();

```

```

if ($hoteldb::status == $hotelc::stat_ok)
{
    $sql_str="UPDATE BILL SET BILL.TOTAL_MEAL = (BILL.TOTAL_MEAL + ";
    $sql_str.=$MEAL_AMOUNT;
    $sql_str.=") where (REGISTRATION_ID=\\";
    $sql_str.=$REGISTRATION_ID;
    $sql_str.=")";
    &_get_db_data($sql_str);
    &_db_close;
}
)

```

sub ADD_LAUND_AMT

```

{
    local ($REGISTRATION_ID,$LAUND_AMOUNT)=@_;

    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="UPDATE BILL SET BILL.TOTAL_LAUNDRY = (BILL.TOTAL_LAUNDRY + ";
        $sql_str.=$LAUND_AMOUNT;
        $sql_str.=") where (REGISTRATION_ID=\\";
        $sql_str.=$REGISTRATION_ID;
        $sql_str.=")";
        &_get_db_data($sql_str);
        &_db_close;
    }
}

```

sub RET_PAYMENT_LIST

```

{
    local($PAYMENT_TYPE,*WA_CARD_COMPANY,*WA_DISCOUNT)=@_;

    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        if ($PAYMENT_TYPE == 3)
        {
            $sql_str="select CORP_NAME,CORP_DISCOUNT from CORPORATE";
        }
        else
        {
            $sql_str="select CARD_COMPANY,DISCOUNT from CARDS";
            $sql_str.=" where (CARD_TYPE= ";
            $sql_str.=$PAYMENT_TYPE;
            $sql_str.=")";
        }
        &_get_db_data($sql_str);
        $i=0;
    }
}

```

```
while ($Data->FetchRow() )
```

```
{
```

```
    @arr = $Data->Data();
```

```
    $WA_CARD_COMPANY[$i] = $arr[0] ;
```

```
    $WA_DISCOUNT[$i] = $arr[1];
```

```
    $i++;
```

```
}
```

```
&_db_close;
```

```
}
```

```
}
```

```
sub UPDATE_PAYMENT
```

```
{
```

```
    local ($REGISTRATION_ID)=@_;
```

```
    $hoteldb::status = _opendb();
```

```
    if ($hoteldb::status == $hotelc::stat_ok)
```

```
    {
```

```
        $sql_str="UPDATE PAYMENT SET PAY_CLOSED = 0";
```

```
        $sql_str.=" where (REGISTRATION_ID=\''";
```

```
        $sql_str=$REGISTRATION_ID;
```

```
        $sql_str='\')";
```

```
        &_get_db_data($sql_str);
```

```
        &_db_close;
```

```
    }
```

```
}
```

```
sub RET_STATE_ROOM
```

```
{
```

```
    local ($ROOM_NO,*ROOM_STATE)=@_;
```

```
    $hoteldb::status = _opendb();
```

```
    if ($hoteldb::status == $hotelc::stat_ok)
```

```
    {
```

```
        $sql_str="select ROOM_STATE from ROOMS";
```

```
        $sql_str.=" where (ROOM_NO=\''";
```

```
        $sql_str=$ROOM_NO;
```

```
        $sql_str='\')";
```

```
        &_get_db_data($sql_str);
```

```
        $status = $Data->FetchRow();
```

```
        $ROOM_STATE = $Data->Data();
```

```
        &_db_close;
```

```
    }
```

```
}
```

```
sub UPDATE_ROOM_STATE
```

```
{
```

```
    local ($ROOM_NO)=@_;
```

```
    &RET_STATE_ROOM($ROOM_NO,$ROOM_STATE);
```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    if ($RM_ST == 1)
    {
        $sql_str="UPDATE ROOMS SET ROOM_STATE = 2";
        $sql_str.=" where (ROOM_NO=\\";
        $sql_str=$ROOM_NO;
        $sql_str.=")";
        &_get_db_data($sql_str);
    }
    if ($RM_ST == 3) {
        $sql_str="UPDATE ROOMS SET ROOM_STATE = 0";
        $sql_str.=" where (ROOM_NO=\\";
        $sql_str=$ROOM_NO;
        $sql_str.=")";
        &_get_db_data($sql_str);
    }
    &_db_close;
}
}

```

```

sub RET_CHK_OUT_ROOM_LIST

```

```

{
    local($DATE,*WA_ROOMS)=@_;

```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    $sql_str="select ROOM_NO from REGISTRATION";
    $sql_str.=" where (ACT_CHECK_OUT = #";
    $sql_str=$DATE;
    $sql_str.=") ORDER BY ROOM_NO";
    &_get_db_data($sql_str);
    $id = 0;

```

```

while ($Data->FetchRow() )

```

```

{
    $WA_ROOMS[$id] = $Data->Data();
    $id++;
}
&_db_close;
}

```

```

sub CHK_VACANCY

```

```

{
    local ($ROOM_NO,*VACANT)=@_;

```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)

```

```

{
    $sql_str="select ROOM_STATE from ROOMS";
    $sql_str=" where (ROOM_NO=\\";
    $sql_str=$ROOM_NO;
    $sql_str="\\";
    &_get_db_data($sql_str);
    $status = $Data->FetchRow();
    @arr = $Data->Data();
    $what = $arr[0];
    if ($what == 1)
    {
        $VACANT = 0;
    }
    else
    {
        $VACANT = 1;
    }
    &_db_close;
}
}

sub RET_EMPTY_ROOM_LIST
(
    local ($ROOM_TYPE,*ROOM_LIST)=@_;

    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="select ROOM_NO from ROOMS";
        $sql_str=" where ((ROOM_TYPE = ";
        $sql_str=$ROOM_TYPE;
        $sql_str=" ) AND ( ROOM_STATE = 2 )) ORDER BY ROOM_NO";
        &_get_db_data($sql_str);
        $id = 0;
        while ($Data->FetchRow() )
        {
            $ROOM_LIST[$id] = $Data->Data();
            $id++;
        }
        &_db_close;
    }
}

sub UPDATE_CANCEL_ROOM_STATE
(
    local (*WA_ROOM_NO)=@_;

    $n=scalar(@WA_ROOM_NO);
    for($idx=0;$idx<$n;$idx++)
    {
        $ROOM_NO = $WA_ROOM_NO[$idx];
        &RET_STATE_ROOM($ROOM_NO,$ROOM_STATE);
        $RSS[$idx] = $ROOM_STATE;
    }
}

```

```

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    for($idx=0;$idx<$n;$idx++)
    {
        $RS = $RSS[$idx];
        if ($RS == 3)
        {
            $sql_str="UPDATE ROOMS SET ROOM_STATE = 0";
            $sql_str.=" where (ROOM_NO=\\";
            $sql_str.=$WA_ROOM_NO[$idx];
            $sql_str.=")";
            &_get_db_data($sql_str);
        }
        if ($RS == 1) {
            $sql_str="UPDATE ROOMS SET ROOM_STATE = 2";
            $sql_str.=" where (ROOM_NO=\\";
            $sql_str.=$WA_ROOM_NO[$idx];
            $sql_str.=")";
            &_get_db_data($sql_str);
        }
    }
    &_db_close;
}

sub RET_ROOM_STATE_LIST
{
    local (*ROOM_TYPE,$DATE,*ROOM_NO,*ROOM_STATE)=@_;
    @ROOM_NO;
    $dsn = new Win32::ODBC("DSN=ntoracle;UID=W10;PWD=W10;");
    if (!$dsn)
    {
        print "create error ";
        die;
    }
    $sqlstr = "select (To_date(\"";
    $sqlstr .= $DATE;
    $sqlstr .= "\", 'mm/dd/yyyy') + 1) from dual";
    $dsn->Sql($sqlstr);
    $dsn->FetchRow();
    $DATE = $dsn->Data();
    $dsn->Close();
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $nnn = scalar(@ROOM_TYPE);

        @ROOM_NO;
    }
}

```

```

@ROOM_STATE;

for($idx=0;$idx<$nnn;$idx++)
{
    $RT=$ROOM_TYPE[$idx];

    $sql_str="select ROOM_NO from ROOMS";
    $sql_str+=" where ((ROOM_TYPE = ";
    $sql_str.=$RT;
    $sql_str.=" ) AND ( ROOM_STATE = 2 )) ORDER BY ROOM_NO";
    &_get_db_data($sql_str);
    $i=0;
    while ($Data->FetchRow() )
    {
        @arr = $Data->Data();
        push @ROOM_NO, $arr[0] ;
        push @ROOM_STATE, 2;
        $i++;
    }
    $sql_str="SELECT ROOMS.ROOM_NO, ";
    $sql_str.=" REGISTRATION.ACT_CHECK_OUT, ";
    $sql_str.="ROOMS.ROOM_STATE,ROOMS.ROOM_TYPE ";
    $sql_str.="FROM REGISTRATION INNER JOIN";
    $sql_str.=" ROOMS ON REGISTRATION.ROOM_NO ";
    $sql_str.="= ROOMS.ROOM_NO WHERE (((REGISTRATION";
    $sql_str.="ACT_CHECK_OUT)=#";
    $sql_str.=$DATE;
    $sql_str.="#) AND ((ROOMS.ROOM_STATE)=1)AND ((ROOMS.ROOM_TYPE)=";
    $sql_str.=$RT;
    $sql_str.=")";
    &_get_db_data($sql_str);
    $i=0;
    while ($Data->FetchRow() )
    {
        @arr = $Data->Data();
        push @ROOM_NO, $arr[0] ;
        push @ROOM_STATE, 1;
        $i++;
    }
}
&_db_close;
}

sub RETURN_ROOM_STATE_LIST2
{
    local (*ROOM_TYPE,$DATE,*ROOM_NO,*ROOM_STATE)=@_;
    @ROOM_NO;
    # $dsn = new Win32::ODBC("DSN=ntoracle;UID=W10;PWD=W10;");
    #if ($dsn)

```

```

# {
# print "create error ";
# die;
# }

# $sqlstr = "select (To_date('";
# $sqlstr .= $DATE;
# $sqlstr .= "','mm/dd/yyyy') + 1) from dual";
# $dsn->Sql($sqlstr);
# $dsn->FetchRow();
# $DATE = $dsn->Data();
# $dsn->Close();

$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    $nnn = scalar(@ROOM_TYPE);

    @ROOM_NO;
    @ROOM_STATE;

    for($idx=0;$idx<$nnn;$idx++)
    {
        $RT=$ROOM_TYPE[$idx];

        $sql_str="select ROOM_NO from ROOMS";
        $sql_str.=" where ((ROOM_TYPE = ";
        $sql_str.=$RT;
        $sql_str.=") AND ( ROOM_STATE = 2 )) ORDER BY ROOM_NO";
        &_get_db_data($sql_str);
        $i=0;
        while ($Data->FetchRow() )
        {
            @arr = $Data->Data();
            push @ROOM_NO, $arr[0] ;
            push @ROOM_STATE, 2;
            $i++;
        }

        $sql_str="SELECT ROOMS.ROOM_NO,";
        $sql_str.=" REGISTRATION.ACT_CHECK_OUT, ";
        $sql_str.="ROOMS.ROOM_STATE,ROOMS.ROOM_TYPE";
        $sql_str.="FROM REGISTRATION INNER JOIN";
        $sql_str.=" ROOMS ON REGISTRATION.ROOM_NO ";
        $sql_str.="= ROOMS.ROOM_NO WHERE (((REGISTRATION";
        $sql_str.="ACT_CHECK_OUT)=#";
        $sql_str.=$DATE;
        $sql_str.=") AND ((ROOMS.ROOM_STATE)=1)AND ((ROOMS.ROOM_TYPE)=";
        $sql_str.=$RT;
        $sql_str.="))";
        &_get_db_data($sql_str);
        $i=0;
        while ($Data->FetchRow() )

```

```

{
    @arr = $Data->Data();
    push @ROOM_NO, $arr[0] ;
    push @ROOM_STATE, 1;
    $i++;
}
}
&_db_close;
}
}

```

```
1;
```

Laundry.pl

```

require "cgi_lib.pl";
use hotelc;
use laundry;
&ReadParse();
@CLOTH_QUANTITY;
@CLOTH_ID;
$LAUND_AMOUNT;

$ROOM_NO = $in{TXROOM_NO};
$LAUND_DATE = $in{TXLAUND_DATE};
$CLOTH_QUANTITY[0] = $in{TX1};
$CLOTH_QUANTITY[1] = $in{TX2};
$CLOTH_QUANTITY[2] = $in{TX3};
$CLOTH_QUANTITY[3] = $in{TX4};
$CLOTH_QUANTITY[4] = $in{TX5};

$CLOTH_ID[0]="001";
$CLOTH_ID[1]="003";
$CLOTH_ID[2]="002";
$CLOTH_ID[3]="004";
$CLOTH_ID[4]="005";

$CLOTH_NAME[0]="T-SHIRTS";
$CLOTH_NAME[1]="COATS";
$CLOTH_NAME[2]="TROUSERS";
$CLOTH_NAME[3]="SUITES";
$CLOTH_NAME[4]="SHIRTS";

&hoteldb::RET_STATE_ROOM($ROOM_NO,$STATUS);
if ( ($STATUS == $hotelc::rm_registered ) ||
($STATUS == $hotelc::rm_asg_checkin ))
{
    &laundry::CAL_LAUND_AMT($ROOM_NO,$LAUND_DATE,@CLOTH_ID,@CLOTH_QUANTITY,$LAUND_AMOUNT);

```

```

&createOK();
}
else
{
    &create_error();
}
sub create_error()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print '<HTML><BODY BGCOLOR=white>';

    print ' <title>LAUNDRY- Update ERROR!</title>';
    print ' <h1><font color="#FF0000" size="6">LAUNDRY-Update ERROR!</font></h1>';
    print ' <center>';
    print ' <table border="1" background=red>';
    print ' <tr>';
    print ' <td><font size="1">ROOM NO. </font></td>';
    print ' <td><font size="1"><input type="text" size="10" name="TXROOM_NO" value="';print $ROOM_NO;print '></font></td>';
    print ' <td><font size="1">DATE </font></td>';
    print ' <td><font size="1"><input type="text" size="13" name="TXLAUNDRY_DATE" value="';print $LAUND_DATE;print '> </font></td>';
    print ' </tr>';
    print ' </table>';
    print ' <br>';
    print ' Room not Occupied';
    print ' <br>';
    print ' <br>';
    print ' <a href="..../prot.htm">Back to Menu</a><br>';
    print ' </center>';
    print ' </html>';
}

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print '<HTML><BODY BGCOLOR=WHITE>';

    print ' <title>LAUNDRY- Update OK!</title>';
    print ' <h1><font color="#FF0000" size="6">LAUNDRY-Update OK!</font></h1>';
    print ' <center>';
    print ' <table border="1" background="..../bg080.jpg">';
    print ' <tr>';
    print ' <td><font size="1">ROOM NO. </font></td>';
    print ' <td><font size="1"><input type="text" size="10" name="TXROOM_NO" value="';print $ROOM_NO;print '></font></td>';
    print ' <td><font size="1">DATE </font></td>';
    print ' <td><font size="1"><input type="text" size="13" name="TXLAUNDRY_DATE" value="';print $LAUND_DATE;print '> </font></td>';
    print ' </tr>';
    print ' </table>';
    print ' <table border="1" background="..../bg080.jpg">';

```

```

print ' <tr>';
print ' <td><font size="1">CLOTH TYPE</font></td>';
print ' <td><font size="1">QUANTITY</font></td>';
print ' </tr>';

for ($i=0;$i<scalar(@CLOTH_ID);$i++)
{
print ' <tr>';
print ' <td><font size="1">;print $CLOTH_NAME;print '</font></td>';
print ' <td><font size="1"><input type="text" size="10" value="';print $CLOTH_QUANTITY;print '> </font></td>';
print ' </tr>';
}

print '</table>';

print '<br>';
print 'Total Price: ';print $LAUND_AMOUNT;
print '<br>';
print '<a href=" ../prot.htm">Back to Menu</a><br>';
print '</center>';
print '</html>';
}

```

Laundry.pm

```

package laundry;
use clothes;
use hoteldb;
my (%data);
sub new
{
    $data{REGISTRATION_ID} = undef;
    $data{LAUND_DATE} = undef;
    $data{LAUND_AMOUNT} = undef;
    $data{CLOTH_ID} = undef;
    $data{CLOTH_QUANTITY} = undef;
    $data{ROOM_NO} = undef;
}

sub CAL_LAUND_AMT
{
    local ($ROOM_NO,$LAUND_DATE,*CLOTH_ID,*CLOTH_QUANTITY,*LAUND_AMOUNT);
    @CLOTH_PRICE;
    &clothes::RET_CLOTH_PRICE(@CLOTH_ID,@CLOTH_PRICE);
    $data{ROOM_NO}=$ROOM_NO;
    $data{LAUND_DATE}=$LAUND_DATE;
    $size_m=scalar(@CLOTH_ID);
    for ($i=0;$i<$size_m;$i++)
    {
        $LAUND_AMOUNT=$LAUND_AMOUNT+($CLOTH_QUANTITY[$i] * $CLOTH_PRICE[$i]);
    }
}

```

```

}

$data{LAUND_AMOUNT} = $LAUND_AMOUNT;
#$data{CLOTH_ID} = $CLOTH_ID;
#$data{CLOTH_QUANTITY} = $CLOTH_QUANTITY;
&hoteldb::UPT_LAUND_AMT(%data);
}

sub UPT_LAUND_AMT
{
local ($ROOM_NO,$LAUND_DATE,$LAUND_AMOUNT)=@_ ;
$REGISTRATION_ID;
$D1;
$D2;
$D3;
$D4;
$PERSON_NAME;
printf "room %s\n",$ROOM_NO;
#-----EN SON BURAYI SILDIM -----NEW();
&registration::RET_REG_ID($ROOM_NO,$REGISTRATION_ID,$D1,$D2,$D3,$D4,$PERSON_NAME);
printf "REG %s\n",$REGISTRATION_ID;
printf "PER %s\n",$PERSON_NAME;

$data{REGISTRATION_ID} = $REGISTRATION_ID;
$data{LAUND_DATE} = $LAUND_DATE;
$data{LAUND_AMOUNT} = $LAUND_AMOUNT;
$data{LAUND_DESC} = $PERSON_NAME;
$data{ROOM_NO} = $ROOM_NO;
&hoteldb::UPT_LAUND_AMT(%data);
}

```

1;

Meal.pm

```

package meal;
use hoteldb;
use registration;
my (%data);
$meal::status;
$meal::err_str;
sub NEW
{
$data{REGISTRATION_ID} = undef;
$data{MEAL_DATE} = undef;
$data{MEAL_AMOUNT} = undef;
$data{MEAL_DESC} = undef;
$data{ROOM_NO} = undef;
}

```

sub UPT_MEAL_AMT

```
{
local ($ROOM_NO,$MEAL_DATE,$MEAL_AMOUNT)=@_ ;
$REGISTRATION_ID;
$D1;
$D2;
$D3;
$D4;
$PERSON_NAME;
#printf "room %s \n",$ROOM_NO;
#-----EN SON BURAYI SILDIM _____NEW();
&registration::RET_REG_ID($ROOM_NO,$REGISTRATION_ID,$D1,$D2,$D3,$D4,$PERSON_NAME);
#printf "REG %s \n",$REGISTRATION_ID;
#printf "PER %s \n",$PERSON_NAME;
```

```
$data{REGISTRATION_ID} = $REGISTRATION_ID;
$data{MEAL_DATE} = $MEAL_DATE;
$data{MEAL_AMOUNT} = $MEAL_AMOUNT;
$data{MEAL_DESC} = $PERSON_NAME;
$data{ROOM_NO} = $ROOM_NO;
&hoteldb::UPT_MEAL_AMT(%data);
}
1;
```

Mealc.pl

```
require "cgi_lib.pl";
&ReadParse;
use hotelc;
use meal;
$STATUS;
$MEAL_DATE=$in{TXMEAL_DATE};
$ROOM_NO=$in{TXROOM_NO};
$MEAL_AMOUNT=$in{TXMEAL_AMOUNT};
&hoteldb::RET_STATE_ROOM($ROOM_NO,$STATUS);
if ( ($STATUS == $hoteldb::rm_registered ) ||
($STATUS == $hotelc::rm_asg_checkin ))
{
&meal::UPT_MEAL_AMT($ROOM_NO,$MEAL_DATE,$MEAL_AMOUNT);

&head();
}
else
{
&onerror();
}

sub head
{
print "HTTP/1.0 200 OK\n";
```

```

print "Content-type: text/html\n\n";
print "<HTML><BODY BGCOLOR=WHITE>";
print " Room no $ROOM_NO ";
print "Updated <P>";
print " date $MEAL_DATE";
print " amount $MEAL_AMOUNT";

print "<A HREF=../prot.htm>Back to menu</A>";
print "</BODY></HTML>";
}

```

```

sub onerror
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print "<HTML><BODY BGCOLOR=WHITE>";
print "the room is $ROOM_NO is not occupied ";
print "<A HREF=../meal.htm>Back to meal</A>";
print "</BODY></HTML>";
}

```

Payment.pm

```

package payment;
use Win32::ODBC;
use hoteldb;

$dsn = new Win32::ODBC("DSN=ntoracle;uid=w9;pwd=W9");
my (%data,$G_PAYMENT_AMOUNT,$G_PAYMENT_TYPE);
sub new
{
    $data{PAYMENT_AMOUNT} = undef;
    $data{PAYMENT_TYPE} = undef;
    $data{CARD_NO} = undef;
    $data{CARD_ID} = undef;
    $data{CORP_ID} = undef;
    $data{REGISTRATION_ID} = undef;
    $data{ROOM_NO} = undef;
}

sub CAL_TOT_AMT
{
    local(*BILL_INFO,*PAYMENT_AMOUNT,*ROOM_CHARGE,*SERVICE_CHARGE,*TAX)=@_;
    $PAYMENT_AMOUNT=$BILL_INFO{TOTAL_MEAL} + $BILL_INFO{TOTAL_LAUNDRY} + $BILL_INFO{LAST_MIN_CHARGE};
    if ($dsn)
    {
        print "create error ";
        die;
    }
}

```

```

$Sqlstr = "select To_date('";
$Sqlstr.= $BILL_INFO(DATE);
$Sqlstr.= '\', 'yyyy/mm/dd') - To_date('";
$Sqlstr.= $BILL_INFO(ACT_CHECK_IN);
$Sqlstr.= '\', 'yyyy/mm/dd') from dual";
$dns->Sql($Sqlstr);
$dns->FetchRow();
$days = $dns->Data();
$dns->Close();
$ROOM_CHARGE = $days * $BILL_INFO(ROOM_RATE);
$SERVICE_CHARGE=1000;
$PAYMENT_AMOUNT=$PAYMENT_AMOUNT+$ROOM_CHARGE+$SERVICE_CHARGE;
$TAX=$PAYMENT_AMOUNT*0.05;
$PAYMENT_AMOUNT=$PAYMENT_AMOUNT+$TAX;
$G_PAYMENT_AMOUNT=$PAYMENT_AMOUNT;
}

sub RET_PAY_TYPE_LIST
{
local($PAYMENT_TYPE,*WA_CARD_COMPANY,*WA_DISCOUNT)=@_;
$G_PAYMENT_TYPE = $PAYMENT_TYPE;
&hoteldb::RET_PAYMENT_LIST($PAYMENT_TYPE,@WA_CARD_COMPANY,@WA_DISCOUNT);
}

sub CAL_NET_TOT_AMT
{
local($CARD_NO,$DISCOUNT,$COMPANY_NAME,*TOTAL_AMOUNT)=@_;
$R_DISC = 100 - $DISCOUNT;
$R_DISC /= 100;
$TOTAL_AMOUNT = $G_PAYMENT_AMOUNT * $R_DISC;
}

sub CLOSE_PAYMENT
{
local($REGISTRATION_ID)=@_;
&hoteldb::UPDATE_PAYMENT($REGISTRATION_ID);
}

1;

```

Registration_list.pl

```

-----
use registration;
use hotelc;

require "cgi_lib.pl";
&ReadParse();

$date = $in{HDDATE};

```

```

#$date = "1997/12/15";
@WA_ROOM_NO;
@WA_REGISTRATION_ID;

&registration::RET_REG_LIST($date,@WA_ROOM_NO,@WA_REGISTRATION_ID);

if ( $registration::status==$hotelc::stat_ok )
{ &createOK(); }
else { &reserror(); }

sub createOK()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<HTML><BODY background= "../bg080.jpg">';

print '<title>Registration List</title>';
print '<p><font color="#FF0080" size="4">Registration List</font></p>';
print '<p>DATE: <input type="text" size="20" name="T1" value="';print $date; print '></p>';
print '<table border="1">';
print ' <tr>';
print ' <td>ROOM NUMBER</td>';
print ' <td>REGISTRATION ID</td>';
print ' </tr>';

for ($i=0;$i<scalar(@WA_ROOM_NO);$i++)
{
print ' <tr>';
print ' <td>';print $WA_ROOM_NO[$i];print '</td>';
print ' <td>';print $WA_REGISTRATION_ID[$i];print '</td>';
print ' </tr>';
}

print '</table>';
print ' <a href=../prot.htm>Back to Menu</a><br>';
print '</body>';
print '</html>';
}

sub reserror()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print '<html>';
print '<head>';
print '<meta http-equiv="Content-Type";';
print 'content="text/html; charset=iso-8859-1">';
print '<meta name="GENERATOR" content="Microsoft FrontPage 2.0">';

```

```

print '<title>Check-out List</title>';
print '</head>';
print '<body bgcolor="#FFFFFF">';
print '<p><font color="#FF0080" size="5">Registered List - No list Produced</font></p>';
print '<p>&nbsp;</p>';
print '<hr>';
print '<p align="center"><a href="/hotel/reg_jst.htm">Back to Reservation</a></p>';
print '</body>';
print '</html>';
}

```

Registration_information.pl

```

require "cgi_lib.pl";
&ReadParse();
$reserv_id = $in{HDRESERVE_ID};

&CreateRegInfo();
sub CreateRegInfo
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print '<html>';
    print '<title>Registration-Get customer Information</title>';
    print '<body background="http://cicc09/hotel/bg080.jpg">';
    print '<h1><font color="#FF0080">Registration-Get customer information</font> </h1>';
    print '<center>';
    print '<table border="1">';
    print '<tr>';
    print '<td>RESERVE ID : </font></td>';
    print '<td><input type="text" size="12" name="TXRESERVE_ID" value="1234567890"></td>';
    print '<td>REGISTER ID :</td>';
    print '<td><input type="text" size="15" name="TXREGISTER_ID" value="1234567890101"></td>';
    print '<td>CHECK IN :<input type="text" size="15" name="TXDATE" value="1997/12/18"></td>';
    print '</tr>';
    print '</table>';
    print '<form action="_yti_bin/shtml.dll/regis_info.htm" method="POST">';
    print '<table border="0">';
    print '<tr>';
    print '<td>ROOM NO.</td>';
    print '<td>&nbsp;</td>';
    print '<td>NAME :</td>';
    print '<td><input type="text" size="30" name="TXPERSON_NAME" value="MR.XXXXXXXXXXXXXXXXXX"></td>';
    print '</tr>';
    print '<tr>';
    print '<td><input type="text" size="10" name="TXROOM_NO" value="101"></td>';
    print '<td>&nbsp;</td>';
    print '<td>PASSPORT NO.</td>';

```

```

print ' <td><input type="text" size="20" name="TXPASSPORT_NO" value="K12312345"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>TYPE</td>';
print ' <td>&nbsp;</td>';
print ' <td>ID NO.</td>';
print ' <td><input type="text" size="20" name="TXID_CARD_NO" value="Z123456789"></td>';
print ' </tr>';
print ' <tr>';
print ' <td><input type="text" size="10" name="TXROOM_TYPE" value="SINGLE"></td>';
print ' <td>&nbsp;</td>';
print ' <td>ADDRESS</td>';
print ' <td><input type="text" size="40" name="TXCONTACT_ADDRESS" value="123 YOKOHAMA KENSHU CENTER., YOKOHAMA"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>PHONE</td>';
print ' <td><input type="text" size="20" name="TX_CONTACT_PHONE" value="054 1234567"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>CHECK OUT</td>';
print ' <td><input type="text" size="20" name="TXACT_CHECK_OUT" value="1997/12/25"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>&nbsp;</td>';
print ' <td>PAYMENT</td>';
print ' <td><select name="PAYMENT_TYPE" size="1">';
print ' <option selected value="CASH">CASH</option>';
print ' <option value="COMPANY">COMPANY</option>';
print ' <option value="CREDIT CARD">CREDIT CARD</option>';
print ' <option value="TRAVEL CHECK">TRAVEL CHECK</option>';
print ' </select></td>';
print ' </tr>';
print ' </table>';
print ' <input type="submit" value="Submit">';
print ' <input type="reset" value="Reset">';
print ' </form>';
print ' </center>';
print ' </body>';
print ' </html>';
}

```

```

require "cgi_lib.pl";
&ReadParse();
$indate = $in{HDDATE};
&regupdateOK;

sub regupdateOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print "<HTML><BODY BGCOLOR=WHITE>";
    print "<title>Registration-Update OK!</title>zz";
    print "<body background='bg080.jpg'>";
    print "<h1><font color='FF0080'>Registration-Update OK!</h1>";
    print "<center>";
    print "<table border='1'>";
    print "    <tr>";
    print "        <td>RESERVE ID : </td>";
    print "        <td><input type='text' size='12' name='TXRESERVE_ID' value='1234567890'></td>";
    print "        <td>REGISTER ID :</td>";
    print "        <td><input type='text' size='15' name='TXREGISTER_ID' value='1234567890101'> </td>";
    print "        <td>CHECK IN :<input type='text' size='15' name='TXDATE' value='1997/12/18'></td>";
    print "    </tr>";
    print "</table>";
    print "<table border='1'>";
    print "    <tr>";
    print "        <td>ROOM NO.</td>";
    print "        <td>NAME :</td>";
    print "        <td><input type='text' size='30' name='TXPERSON_NAME' value='MR.XXXXXXXXXXXXXXXXXX'></td>";
    print "    </tr>";
    print "    <tr>";
    print "        <td><input type='text' size='10' name='TXROOM_NO' value='101'></td>";
    print "        <td>PASSPORT NO.</td>";
    print "        <td><input type='text' size='20' name='TXPASSPORT_NO' value='K12312345'></td>";
    print "    </tr>";
    print "    <tr>";
    print "        <td>TYPE</td>";
    print "        <td>ID NO.</td>";
    print "        <td><input type='text' size='20' name='TXID_CARD_NO' value='Z123456789'></td>";
    print "    </tr>";
    print "    <tr>";
    print "        <td><input type='text' size='10' name='TXROOM_TYPE' value='SINGLE'></td>";
    print "        <td>ADDRESS</td>";
    print "        <td><input type='text' size='40' name='TXCONTACT_ADDRESS' value='123 YOKOHAMA KENSHU CENTER., YOKOHAMA'></td>";
    print "    </tr>";
    print "    <tr>";
    print "        <td>&nbsp;</td>";
    print "        <td>PHONE</td>";
    print "        <td><input type='text' size='20' name='TX_CONTACT_PHONE' value='054 1234567'></td>";

```

```

print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>CHECK OUT</td>';
print ' <td><input type="text" size="20" name="TXACT_CHECK_OUT" value="1997/12/25"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>&nbsp;</td>';
print ' <td>PAYMENT</td>';
print ' <td><input type="text" size="20" name="TXPAYMENT_TYPE" value="CASH"></td>';
print ' </tr>';
print ' </table>';
print ' <p align="center"><a href="mainmenu.htm">Back to Menu</a></p>';
print '<hr>';
print '</body>';
print '</html>';
}

```

Registration_report.pl

```

require "cgi_lib.pl";
&ReadParse();
$date = $in{HDDATE};
&createOK;

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print '<HTML><BODY BGCOLOR=WHITE>';
    print '<title>REGISTER-Report</title>';
    print '<body background="http://cicc09/hotel/bg080.jpg">';
    print '<font color="#FF0080" size="4">REGISTRATION-Report at the end of the day</font></p>';
    print '<p>DATE : <input type="text" size="20" name="T1" value=;
print "$date";
print '></p>';
print '<table border="1">';
print ' <tr>';
print ' <td>REGISTER ID</td>';
print ' <td>ROOM TYPE</td>';
print ' <td>ROOM NO.</td>';
print ' <td>CUSTOMER NAME</td>';
print ' <td>SEX</td>';
print ' <td>CHECK IN</td>';
print ' <td>CHECK OUT</td>';
print ' <td>COMPANY</td>';
print ' </tr>';
print ' <tr>';

```



```

print '<font size="1">BACK</font>';
print '</body>';
print '</html>';
}

```

Registration.pm

```
package registration;
```

```
use hoteldb;
```

```
use hotelc;
```

```
$registration::status;
```

```
$registration::err_str;
```

```
my (%data);
```

```
sub new
```

```

{
    $data{REGISTRATION_ID}=undef;
    $data{PERSON_NAME}=undef;
    $data{RESERVE_ID}=undef;
    $data{PASSPORT_NO}=undef;
    $data{ID_CARD_NO}=undef;
    $data{CONTACT_ADDR}=undef;
    $data{CONTACT_PHONE}=undef;
    $data{ROOM_NO}=undef;
    $data{ACT_CHECK_IN}=undef;
    $data{ACT_CHECK_OUT}=undef;
    $data{PRE_REG_ID}=undef;
    $data{PAYMENT_TYPE}=undef;
    $data{REG_CLOSED}=undef;
    $data{ROOM_TYPE}=undef;
    $data{ROOM_RATE}=undef;
}

```

```
sub RET_REG_ID
```

```
#tested
```

```

{
    local ($ROOM_NO,*REGISTRATION_ID,*ROOM_TYPE,*ACT_CHECK_IN,*PAYMENT_TYPE,*ROOM_RATE,*PERSON_NAME)=@_;

```

```

    &hoteldb::RET_REG_ID($ROOM_NO,$REGISTRATION_ID,$ROOM_TYPE,$ACT_CHECK_IN,$PAYMENT_TYPE,$ROOM_RATE,$PERSON_NAME);
}

```

```
$data{ROOM_NO}=$ROOM_NO;
```

```
$data{REGISTRATION_ID}=$REGISTRATION_ID;
```

```
$data{ROOM_TYPE}=$ROOM_TYPE;
```

```
$data{ACK_CHECK_IN}=$ACK_CHECK_IN;
```

```
$data{PAYMENT_TYPE}=$PAYMENT_TYPE;
```

```
$data{ROOM_RATE}=$ROOM_RATE;
```

```
$data{PERSON_NAME}=$PERSON_NAME;
```

```

}

#end sub RET_REG_ID

sub RET_ACT_CHK_OUT_LIST
# tested
{
    local ($DATE,*WA_ROOM_NO)=@_;
    &hoteldb::RET_ACT_CHK_OUT_LIST($DATE,@WA_ROOM_NO);
    $registration::status=$hoteldb::status;
    $registration::err_str=$hoteldb::err_str;
}

#end sub RET_ACT_CHK_OUT_LIST

sub ASIGN_ROOM_NO
#tested
{
    local ($RESERVE_ID,*WA_ROOM_NO,*WA_ROOM_STATE)=@_;
    $length=scalar(@WA_ROOM_NO);

    for ($i=0;$i<$length;$i++)
    {
        if ($WA_ROOM_STATE[$i]==$hoteldb::rm_empty)
        {
            $WA_ROOM_STATE[$i]=$hoteldb::rm_assigned;
        }
        if ($WA_ROOM_STATE[$i]==$hoteldb::rm_registered)
        {
            $WA_ROOM_STATE[$i]=$hoteldb::rm_asg_checkin;
        }
    }

    &hoteldb::CREATE_REGISTRATION($RESERVE_ID,@WA_ROOM_NO,@WA_ROOM_STATE);

    $registration::status=$hoteldb::status;
    $registration::err_str=$hoteldb::err_str;
}

#end sub ASIGN_ROOM_NO

sub RET_RESERVEO_ROOMS
#tested
{
    local ($RESERVE_ID,*WA_ROOM_NO,*WA_ROOM_TYPE)=@_;
    &hoteldb::RET_RESERVED_ROOMS($RESERVE_ID,@WA_ROOM_NO,@WA_ROOM_TYPE);

    $data{RESERVE_ID}=$RESERVE_ID;

    $registration::status=$hoteldb::status;

```

```

$registration::err_str=$hoteldb::err_str;
}

#end sub RET_RESERVED_ROOM


sub UPDATE_REGISTRATION
#tested
{
local (*CUST_INFO,$ROOM_NO,$PAYMENT_TYPE,*REGISTRATION_ID)=@_;

$RESERVE_ID=$data(RESERVE_ID);

&hoteldb::UPDATE_REGISTRATION($RESERVE_ID,%CUST_INFO,$ROOM_NO,$PAYMENT_TYPE,$REGISTRATION_ID);

$data(PERSON_NAME)=$CUST_INFO{PERSON_NAME};
$data(PASSPORT_NO)=$CUST_INFO{PASSPORT_NO};
$data(ID_CARD_NO)=$CUST_INFO{ID_CARD_NO};
$data(CONTACT_ADDRESS)=$CUST_INFO{CONTACT_ADDRESS};
$data(CONTACT_PHONE)=$CUST_INFO{CONTACT_PHONE};
$data(ACT_CHECK_IN)=$CUST_INFO{ACT_CHECK_IN};
$data(ACT_CHECK_OUT)=$CUST_INFO{ACT_CHECK_OUT};
$data(ROOM_NO)=$ROOM_NO;
$data(PAYMENT_TYPE)=$PAYMENT_TYPE;
$data(REGISTRATION_ID)=$REGISTRATION_ID;

$registration::status=$hoteldb::status;
$registration::err_str=$hoteldb::err_str;
}

#end sub UPDATE_REGISTRATION


sub RET_REG_LIST
#tested
{
local ($DATE,*WA_ROOM_NO,*WA_REGISTRATION_ID)=@_;
&hoteldb::RET_REG_LIST($DATE,%WA_ROOM_NO,%WA_REGISTRATION_ID);

$registration::status=$hoteldb::status;
$registration::err_str=$hoteldb::err_str;
}

#end sub RET_REG_LIST


sub CLOSE_REGISTRATION
#tested
{
local ($REGISTRATION_ID)=@_;
&hoteldb::CLOSE_REGISTRATION($REGISTRATION_ID);

$registration::status=$hoteldb::status;

```

```

$registration::err_str=$hoteldb::err_str;
}
#end sub CLOSE_REGISTRATION

```

```

sub CANCEL_REGISTRATION
{
    local ($RESERVE_ID)=@_;
    &hoteldb::CANCEL_REGISTRATION($RESERVE_ID);

    $registration::status=$hoteldb::status;
    $registration::err_str=$hoteldb::err_str;
}
#end sub CANCEL_REGISTRATION

```

```

sub _debug
{
    printf "\n--Debug Information--\n\n";
    printf "Registration ID: %s\n",$data{REGISTRATION_ID};
    printf "Person name: %s\n",$data{PERSON_NAME};
    printf "Reservation ID: %s\n",$data{RESERVE_ID};
    printf "Passport no: %s\n",$data{PASSPORT_NO};
    printf "ID card no: %s\n",$data{ID_CARD_NO};
    printf "Contact address: %s\n",$data{CONTACT_ADDR};
    printf "Contact phone: %s\n",$data{CONTACT_PHONE};
    printf "Room no: %s\n",$data{ROOM_NO};
    printf "Room rate: %s\n",$data{ROOM_RATE};
    printf "Ack check in: %s\n",$data{ACT_CHECK_IN};
    printf "Ack check out: %s\n",$data{ACT_CHECK_OUT};
    printf "Pre reg ID: %s\n",$data{PRE_REG_ID};
    printf "Payment type: %s\n",$data{PAYMENT_TYPE};
    printf "Reg closed: %s\n",$data{REG_CLOSED};
    printf "\n--end of Debug Information--\n\n";
}

```

```
1;
```

Reservation_detail.pm

```

package res_detail;
use hoteldb;
use hotelc;
my (%data,@G_NO_OF_ROOMS,@G_ROOM_TYPE,@GW_ROOM_RATE);

```

```

sub new
{
    $data{RESERVE_ID} = undef;
    $data{NO_OF_ROOMS} = undef;
    $data{ROOM_TYPE} = undef;
}

```

```

}

sub SET_ROOM_TYPE
{
local (*ROOM_TYPE,*NO_OF_ROOMS,*WA_ROOM_RATE)=@_;
@G_NO_OF_ROOMS = @NO_OF_ROOMS;
@G_ROOM_TYPE=@ROOM_TYPE;
@GW_ROOM_RATE=@WA_ROOM_RATE;
$no_of_room_type=scalar(@ROOM_TYPE);
for ($i=0;$i<$no_of_room_type;$i++)
{
$data{NO_OF_ROOMS}=$NO_OF_ROOMS[$i];
$data{ROOM_TYPE}=$ROOM_TYPE[$i];
$data{W_ROOM_RATE}=$WA_ROOM_RATE[$i];
# printf "ROOM type detail (%s): %s\n",$i,$ROOM_TYPE[$i];
# printf "no of detail rooms (%s): %s\n",$i,$NO_OF_ROOMS[$i];
# printf "room rate detail (%s): %s\n",$i,$WA_ROOM_RATE[$i];
}
}

sub CAL_DISC_RATE
{
local (*ROOM_TYPE,$CORP_DISCOUNT,*DISCOUNT_RATE)=@_;
printf "room type %s \n",$ROOM_TYPE[2];
$no_of_type=scalar(@ROOM_TYPE);
for ($i=0;$i<$no_of_type;$i++)
{
if ($ROOM_TYPE[$i]==$hotelc::single)
{
$DISCOUNT_RATE[$i]=$CORP_DISCOUNT * 120
}
elsif
($ROOM_TYPE[$i]==$hotelc::twin)
{
$DISCOUNT_RATE[$i]=$CORP_DISCOUNT * 230
}
elsif
($ROOM_TYPE[$i]==$hotelc::suite)
{
$DISCOUNT_RATE[$i]=$CORP_DISCOUNT * 800
}
}
}

sub UPDATE_RESERVE_DETAIL
{
local ($RESERVE_ID,*ROOM_TYPE,*NO_OF_ROOMS)=@_;
$data{RESERVE_ID}=$RESERVE_ID;
&hoteldb::UPDATE_RESERVE_DETAIL($RESERVE_ID,@ROOM_TYPE,@NO_OF_ROOMS);
}

sub RET_ROOM_DETAIL
{
# INPUT ARRAY OF RESERVE_ID
# OUTPUT ARRAY OF NO_OF_ROOMS,ROOM_TYPE
local (*RESERVE_ID,*NO_OF_SINGLE,*NO_OF_TWIN,*NO_OF_SUITE)=@_;

```

```

@WRESERVE_ID = @RESERVE_ID;
@WNO_OF_SINGLE;
@WNO_OF_TWIN;
@WNO_OF_SUITE;
# print "res_id $WRESERVE_ID[0] \n";
&hoteldb::RET_ROOM_DETAIL(@WRESERVE_ID,@WNO_OF_SINGLE,@WNO_OF_TWIN,@WNO_OF_SUITE);
@NO_OF_SINGLE = @WNO_OF_SINGLE;
@NO_OF_TWIN = @WNO_OF_TWIN;
@NO_OF_SUITE = @WNO_OF_SUITE;

}
1;

```

Reservation.pl

```

use reservation;
use hotelc;

require "cgi_lib.pl";
&ReadParse();

$indate = $in{HDCHECKIN};
$outdate = $in{HDCHECKOUT};
$single = $in{TXSINGLE};
$twin = $in{TXTWIN};
$suite = $in{TXSUITE};

@ROOM_TYPE;
@NO_OF_ROOMS;
@WA_AVAIL_NO_OF_ROOMS;
@WA_ROOM_RATE;
@ACCEPTABLE;

$ROOM_TYPE[0]=$hotelc::single;
$ROOM_TYPE[1]=$hotelc::twin;
$ROOM_TYPE[2]=$hotelc::suite;

$NO_OF_ROOMS[0]=$single;
$NO_OF_ROOMS[1]=$twin;
$NO_OF_ROOMS[2]=$suite;

&reservation::RET_AVAIL_ROOMS($indate,$outdate,@ROOM_TYPE,@NO_OF_ROOMS,@WA_AVAIL_NO_OF_ROOMS,@WA_ROOM_RATE,@
ACCEPTABLE);

if ( ($ACCEPTABLE[0]==1) &&($ACCEPTABLE[1]==1) &&($ACCEPTABLE[2]==1) )
{ &createNext(); }
else { &reserror(); }
#&createOK();

```

```

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";
    print '<HTML><BODY BGCOLOR=WHITE>';
    print "Year ";
    if ( $indate == undef )
    {
        print " undef ";
    }
    else
    {
        print $indate;
        print "<P>";
        print $outdate;
    }
    if ( $single != undef )
    {
        printf "<P>single $single<P>";
    }
    if ( $twin != undef )
    {
        printf "twin $twin<P>";
    }
    if ( $suite != undef )
    {
        printf "suite $suite<P>";
    }
    print "</BODY></HTML>";
}

sub createNext
{
    print "HTTP/1.0 200 OK\n";
    print "Content-Type: text/html\n\n";
    print "<html>";
    print "<body background='./bg080.jpg' bgcolor='#FFFFFF'>";
    print "<h2>Reservation</h2>";
    print "<hr>";
    print "<p>The additional reservation information\n";
    print "<script language=JavaScript>\n";
    print "<!--\n";
    print "function FrontPage_Form1_Validator(theForm)\n";
    print "{\n";
    print "    if (theForm.txname.value == \"\");\n";
    print "\n{\n";
    print "    alert(\"Please enter a value for the \"txname\" field.\");";
    print "\n";
    print "    theForm.txname.focus();";
    print "\n";
}

```

```

print ' return (false);';
print "\n\n";
print ' if (theForm.txname.value.length < 1)';
print "\n";
print '{';
print "\n";
print ' alert("Please enter at least 1 characters in the \'txname\' field.");';
print "\n";
print ' theForm.txname.focus();';
print "\n";
print ' return (false);';
print "\n";
print ' }';
print "\n";
print ' if (theForm.txname.value.length > 40)';
print "\n";
print '{';
print "\n";
print ' alert("Please enter at most 40 characters in the \'txname\' field.");';
print "\n";
print ' theForm.txname.focus();';
print "\n";
print ' return (false);';
print "\n";
print ' }';
print "\n";
print ' if (theForm.txaddress.value == "")';
print '{';
print ' alert("Please enter a value for the \'txaddress\' field.");';
print "\n";
print ' theForm.txaddress.focus();';
print "\n";
print ' return (false);';
print "\n";
print ' }';
print "\n";
print ' if (theForm.txaddress.value.length < 1)';
print "\n";
print '{';
print "\n";
print ' alert("Please enter at least 1 characters in the \'txaddress\' field.");';
print "\n";
print ' theForm.txaddress.focus();';
print "\n";
print ' return (false);';
print "\n";
print ' }';
print "\n";
print ' if (theForm.txaddress.value.length > 40)';
print "\n";

```

```

print '{';
print "\n";
print ' alert("Please enter at most 40 characters in the \''txaddress\' field.");';
print "\n";
print ' theForm.txaddress.focus();';
print "\n";
print ' return (false);';
print "\n";
print '>';
print "\n";
print ' if (theForm.txphone.value == "")';
print "\n";
print '{';
print "\n";
print ' alert("Please enter a value for the \''txphone\' field.");';
print "\n";
print ' theForm.txphone.focus();';
print "\n";
print ' return (false);';
print "\n";
print '>';
print "\n";
print ' if (theForm.txphone.value.length < 1)';
print "\n";
print '{';
print "\n";
print ' alert("Please enter at least 1 characters in the \''txphone\' field.");';
print "\n";
print ' theForm.txphone.focus();';
print "\n";
print ' return (false);';
print "\n";
print '>';
print "\n";
print ' if (theForm.txphone.value.length > 20)';
print "\n";
print '{';
print "\n";
print ' alert("Please enter at most 20 characters in the \''txphone\' field.");';
print "\n";
print ' theForm.txphone.focus();';
print "\n";
print ' return (false);';
print "\n";
print '>';
print "\n";
print ' return (true);';
print "\n";
print '>';
print "\n";

```

```

print ' -->';
print "</script>\n";
print ' <form action="ret_resv_id.pl" method="POST" onsubmit="return FrontPage_Form1_Validator(this)" name="FrontPage_Form1">';
print "\n";
print "<input type=hidden name=hddatein value=$indate>";
print "<input type=hidden name=hddateout value=$outdate>";
print "<input type=hidden name=hdsingle value=$single>";
print "<input type=hidden name=hdtwin value=$twin>";
print "<input type=hidden name=hdsuite value=$suite>";

print "<input type=hidden name=hdavl_room0 value=$WA_AVAIL_NO_OF_ROOMS[0]>";
print "<input type=hidden name=hdavl_room1 value=$WA_AVAIL_NO_OF_ROOMS[1]>";
print "<input type=hidden name=hdavl_room2 value=$WA_AVAIL_NO_OF_ROOMS[2]>";

print "<input type=hidden name=hdrate0 value=$WA_ROOM_RATE[0]>";
print "<input type=hidden name=hdrate1 value=$WA_ROOM_RATE[1]>";
print "<input type=hidden name=hdrate2 value=$WA_ROOM_RATE[2]>";

print ' <div align="center"><center><table border="0">';
print ' <tr>';
print ' <td>Expected date in</td>';
print "<td> $indate</td>";
print ' </tr>';
print ' <tr>';
print ' <td>Expected date out</td>';
print "<td>$outdate</td>";
print ' </tr>';
print ' <tr>';
print ' <td>Single</td>';
print "<td>$single</td>";
print ' </tr>';
print ' <tr>';
print ' <td>Twin</td>';
print "<td>$twin</td>";
print ' </tr>';
print ' <tr>';
print ' <td>Suite</td>';
print "<td>$suite</td>";
print ' </tr>';
print ' </table>';
print ' </center></div><hr>';
print ' <table border="0">';
print ' <tr>';
print ' <td>Name</td>';
print ' <td><!--webbot bot="Validation" s-data-type="String" b-value-required="TRUE" i-minimum-length="1" i-maximum-length="40" -->';
print ' <input type="text" size="40" maxlength="40" name="xname"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>Address</td>';
print ' <td><!--webbot bot="Validation" s-data-type="String" b-value-required="TRUE" i-minimum-length="1" i-maximum-length="40" -->';

```

```

print ' <input type="text" size="40" maxlength="40" name="baddress"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>Phone</td>';
print ' <td><!--webbot bot="Validation" s-data-type="String" b-value-required="TRUE" i-minimum-length="1" i-maximum-length="20" -->';
print ' <input type="text" size="20" maxlength="20" name="bphone"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>Fax</td>';
print ' <td><!--webbot bot="Validation" s-data-type="String" b-value-required="TRUE" i-minimum-length="1" i-maximum-length="20" -->';
print ' <input type="text" size="20" maxlength="20" name="bfax"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>Sex</td>';
print ' <td>Male<input type="radio" checked name="opsex" value="male"> ';
print ' Female<input type="radio" name="opsex" value="female"></td>';
print ' </tr>';
print ' <tr>';
print ' <td>Corp id</td>';
print ' <td>';
print ' <select name="bcorpname" size="1">';
print ' <option value="NONE">None</option>';
print ' <option value="NEC">NEC-12</option>';
print ' <option value="IBM">IBM-10</option>';
print ' <option value="SONY">SONY-8</option>';
print ' <option value="CANON">CANNON-9</option>';
print ' <option value="CICC">CICC-20</option>';
print ' <option value="HITACHI">HITACHI-10</option>';
print ' </select>';
#<input type="text" size="20" name="bcorpid"></td>';

print ' </tr>';
print ' </table>';
print ' <p><input type="submit" value="Submit Form"> ';
print ' <input type="reset" value="Reset Form"> </p>';
print ' </form>';
print ' </body>';
print ' </html>';
}

sub reserror()
{
print "HTTP/1.0 200 OK\n";
print "Content-type: text/html\n\n";
print ' <html>';
print ' <head>';
print ' <meta http-equiv="Content-Type">';
print ' content="text/html; charset=iso-8859-1">';
print ' <meta name="GENERATOR" content="Microsoft FrontPage 2.0">';
print ' <title>RESERVATION-</title>';

```

```
print '</head>';
print '<body bgcolor="#FFFFFF">';
print '<p><font color="#FF0080" size="5">RESERVATION-Reserve room grater than available rooms</font></p>';
print '<p>&nbsp;</p>';
print '<hr>';
print '<p align="center"><a href="/hotel/reservation.htm">Back to Reservation</a></p>';
print '</body>';
print '</html>';
}
```

Reservation.pm

```
package reservation;
use res_detail;
use hoteldb;
use hotelc;
use registration;
use rooms;

$reservation::status;
$reservation::err_str;

my (%data,@RESERVE_ID,@g_room_type,@g_no_of_rooms,@gwa_avail_no_rooms,@g_room_rate);
sub new
{
    $data{RESERVE_ID}=undef;
    $data{CORP_NAME}=undef;
    $data{EXP_CHECK_IN}=undef;
    $data{EXP_CHECK_OUT}=undef;
    $data{CUST_NAME}=undef;
    $data{CUST_ADDR}=undef;
    $data{CUST_SEX}=undef;
    $data{CUST_TEL}=undef;
    $data{CUST_FAX}=undef;
    $data{RES_STATE}=undef;
}

sub RET_AVAIL_ROOMS
{
    # room_type and no_of_rooms are inp(array)

    local($EXP_CHECK_IN,$EXP_CHECK_OUT,*ROOM_TYPE,*NO_OF_ROOMS,*WA_AVAIL_NO_OF_ROOMS,*WA_ROOM_RATE,*ACCEPTABLE)=
    @_;

    $data{EXP_CHECK_IN}=$EXP_CHECK_IN;
    $data{EXP_CHECK_OUT}=$EXP_CHECK_OUT;
    &hoteldb::CHECK_AVAIL_ROOMS($EXP_CHECK_IN,$EXP_CHECK_OUT,@ROOM_TYPE,@NO_OF_ROOMS,@WA_AVAIL_NO_OF_ROOMS,@
    WA_ROOM_RATE);
}
```

```

#i=0;
#printf "check in date %s\n",$EXP_CHECK_IN;
#printf "check out date %s\n",$EXP_CHECK_OUT;
#printf "room type %s\n",$ROOM_TYPE[$i];
#printf "avail no of rooms=%s\n",$WA_AVAIL_NO_OF_ROOMS[$i];
$no_of_room_type=scalar(@ROOM_TYPE);
for ($i=0;$i<$no_of_room_type;$i++)
{

if ($NO_OF_ROOMS[$i] < $WA_AVAIL_NO_OF_ROOMS[$i])
    {$ACCEPTABLE[$i]=1}
else
    {$ACCEPTABLE[$i]=0}
}

@g_room_type=@ROOM_TYPE;
@g_no_of_rooms=@NO_OF_ROOMS;
@gwa_avail_no_rooms=@WA_AVAIL_NO_OF_ROOMS;
@g_room_rate=@WA_ROOM_RATE;
&res_detail::SET_ROOM_TYPE(@ROOM_TYPE,@NO_OF_ROOMS,@WA_ROOM_RATE);

$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;

}

sub RET_DISCOUNT
{
local ($CORP_NAME,*CORP_DISCOUNT)=@_;
$data{CORP_NAME}=$CORP_NAME;
&hoteldb::RET_DISCOUNT($CORP_NAME,$CORP_DISCOUNT);
$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}

sub RET_RESV_ID
{
local (*RESV_INFO,*RESERVE_ID)=@_;
$data{CUST_NAME}=$RESV_INFO{CUST_NAME};
$data{CUST_ADDR}=$RESV_INFO{CUST_ADDR};
$data{CUST_TEL}=$RESV_INFO{CUST_TEL};
$data{CUST_FAX}=$RESV_INFO{CUST_FAX};
$data{CUST_SEX}=$RESV_INFO{CUST_SEX};
$data{RES_STATE}=$hoteldb::res_reserved;
#FOLLOWING ADDED BY MSDF
$data{EXP_CHECK_IN}=$RESV_INFO{EXP_CHECK_IN};
$data{EXP_CHECK_OUT}=$RESV_INFO{EXP_CHECK_OUT};
$data{CORP_NAME}=$RESV_INFO{CORP_NAME};

&hoteldb::RET_INIT_RES_ID($RESERVE_ID);
if ($RESERVE_ID==~/0/)

```

```

{$RESERVE_ID="1"}
else
{$RESERVE_ID += 1};

$data{RESERVE_ID}=$RESERVE_ID;

#shifted the following to control program
#&res_detail::UPDATE_RESERVE_DETAIL($RESERVE_ID);

&hoteldb::UPDATE_RESERVATION(%data);
$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}

sub RET_RESERVE_ID_LIST
{
#RESERVE_ID is array

local ($DATE,*WRESERVE_ID)=@_;

&hoteldb::RET_RESERVE_ID($DATE,%WRESERVE_ID);

# print " reservation $WRESERVE_ID[0] \n";
# print " reservation $WRESERVE_ID[1] \n";
$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}

sub SET_RES_STATE_ASSIGNED
{
&hoteldb::SET_RES_STATE_ASSIGNED($RESERVE_ID);
$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}

sub GET_EXP_CHECK_IN
{
local ($RESERVE_ID,*wdata)=@_;
&hoteldb::GET_EXP_CHECK_IN($RESERVE_ID,%wdata);
$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}

sub CANCEL_RESERVATION
{
local ($RESERVE_ID,*CANCELLED,*RES_STATE,*WA_ROOM_NO,*WA_ROOM_TYPE)=@_;

&hoteldb::SET_RES_STATE_CANCELLED($RESERVE_ID,$RES_STATE);
if ($RES_STATE==$hoteldb::rm_assigned)
{
&registration::RET_RESERVED_ROOMS($RESERVE_ID,%WA_ROOM_NO,%WA_ROOM_TYPE);
&rooms::UPDATE_ROOM_STATE(%WA_ROOM_NO);
&registration::CANCEL_REGISTRATION($RESERVE_ID);

```

```

$reservation::status=$hoteldb::status;
$reservation::err_str=$hoteldb::err_str;
}
}
1;

```

Reserve_error.pl

```

&reserror();
sub reserror()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n\n";

    print '<html>';

    print '<head>';

    print '<meta http-equiv="Content-Type";';

    print 'content="text/html; charset=iso-8859-1">';
    print '<meta name="GENERATOR" content="Microsoft FrontPage 2.0">';
    print '<title>RESERVATION-</title>';
    print '</head>';

    print '<body bgcolor="#FFFFFF">';

    print '<p><font color="#FF0080" size="5">RESERVATION-Reserve room grater than available rooms</font></p>';

    print '<p>&nbsp;</p>';

    print '<hr>';

    print '<p align="center"><a href="reserva1.htm">Back to Reservation</a></p>';
    print '</body>';
    print '</html>';
}

```

Reservf.pm

```

package hoteldb;

use Win32::ODBC;
my ($Data);
$hoteldb::status;

```

```

$hoteldb::err_str;

sub _opendb
{
# $Data = new Win32::ODBC("DSN=ntoracle;UID=W10;PWD=W10;");
$Data = new Win32::ODBC("hotel");
if ( $Data )
{
return ($hotelc::stat_ok);
}
else
{
$hoteldb::err_str="DB Open error\n";
return($hotelc::stat_err);
}
}
#end sub _opendb

sub _get_db_data
{
local($sql_str)=@_;
printf "SQL string : %s\n",$sql_str;
$src = $Data->Sql($sql_str);

if ( $src )
{
$hoteldb::err_str = $error[0];
$hoteldb::err_str .= $error[1];
$hoteldb::err_str .= $error[2];
$Data->Close();
$hoteldb::status=$hotelc::stat_err;
}
}
#end _get_db_data

sub _db_close
{
$Data->Close();
}

sub CHECK_AVAIL_ROOMS
{
local ($EXP_CHECK_IN,$EXP_CHECK_OUT,*ROOM_TYPE,*NO_OF_ROOMS,*WA_AVAIL_NO_OF_ROOMS,*WA_ROOM_RATE)=@_;
$hoteldb::status = _opendb();

if ($hoteldb::status == $hotelc::stat_ok)
{
$no_of_room_type=scalar(@ROOM_TYPE);
for ($i=0;$i<$no_of_room_type;$i++)
{
if ($ROOM_TYPE[$i]==$hotelc::single)

```

```

    { }
elseif
    ($ROOM_TYPE[$i]==$hotelc::twin)
    { }
elseif
    ($ROOM_TYPE[$i]==$hotelc::suite)
    { }
}

SELECT Sum(NO_OF_ROOMS)
FROM RESERVATION , RESERVE_DETAIL
ON RESERVATION.RESERVE_ID = RESERVE_DETAIL.RESERVE_ID
WHERE (((RESERVATION.EXP_CHECK_IN)<=[WEXP_CHECK_IN]) AND
((([WEXP_CHECK_IN])<[reservation].[exp_check_out]) AND
((RESERVE_DETAIL.ROOM_TYPE)=[wtype])) OR
(((RESERVATION.EXP_CHECK_IN)<=[WEXP_CHECK_OUT]) AND
((([WEXP_CHECK_OUT])<[reservation].[exp_check_out]) AND
RESERVATION.RES_STATE NOT=2;
);

```

```

    $sql_str="SELECT ROOMS.ROOM_RATE FROM RESERVATION,ROOMS";
    $sql_str=" WHERE (((REGISTRATION.ACT_CHECK_OUT)=",
    $sql_str=$DATE;
    $sql_str=")");
    &_get_db_data($sql_str);
}

```

```

sub RET_DISCOUNT
(
local ($CORP_NAME,*CORP_DISCOUNT)=@;
$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{

$sql_str="SELECT CORP_DISCOUNT FROM CORPORATE";
    $sql_str=" WHERE CORP_NAME=";
    $sql_str=$CORP_NAME;
    $sql_str="";

    &_get_db_data($sql_str);
while ($Data->FetchRow())
{
    $CORP_DISCOUNT =$Data->Data();
}
}

```

```

    &_db_close;
}
}

sub RET_INIT_RES_ID
{
    local (*RESERVE_ID)=@;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT MAX(RESERVE_ID) FROM RESKEY";

        &_get_db_data($sql_str);
        while ($Data->FetchRow())
        {
            $RESERVE_ID=$Data->Data();
        }

        &_db_close;
    }
}

sub UPDATE_RESERVATION
{
    local (*data)=@;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="INSERT INTO RESERVATION VALUES(";
        $sql_str=$data{RESERVE_ID};
        $sql_str.=",";
        $sql_str=$data{CORP_NAME};
        $sql_str.=",";
        $sql_str=$data{EXP_CHECK_IN};
        $sql_str.=",";
        $sql_str=$data{EXP_CHECK_OUT};
        $sql_str=$data{CUST_NAME};
        $sql_str.=",";
        $sql_str=$data{CUST_ADDR};
        $sql_str.=",";
        $sql_str=$data{CUST_SEX};
        $sql_str.=",";
        $sql_str=$data{CUST_TEL};
        $sql_str.=",";
        $sql_str=$data{CUST_FAX};
        $sql_str.=",";
        $sql_str=$data{RES_STATE};
        $sql_str.=")";

        &_get_db_data($sql_str);
    }
}

```

```

        &_db_close;
    }
}

sub UPDATE_RESKEY
{
    local ($data)=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="
sub RET_RESERVE_ID
{
    local ($DATE,*RESERVE_ID)=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT RESERVATION.RESERVE_ID FROM RESERVATION";
        $sql_str.=" WHERE (((RESERVATION.EXP_CHECK_OUT)='";
        $sql_str=$DATE;
        $sql_str.=")')";

        &_get_db_data($sql_str);

        $i=0;
        while ($Data->FetchRow())
        {
            $RESERVE_ID[$i] = $Data->Data();
            $i++;
        }

        &_db_close;
    }
}

sub SET_RES_STATE_ASSIGNED
local (*RESERVE_ID)=@_;
$hoteldb::status = _opendb();
if ($hoteldb::status == $hotelc::stat_ok)
{
    $no_size=scalar(@RESERVE_ID);
    for ($i=0;$i<$no_size;$i++)
    {
        $sql_str="update RESERVATION set RES_STATE=";
        $sql_str=$hotelc::res_reserved;
        $sql_str.="where RESERVE_ID=";
        $sql_str=$RESERVE_ID[$i];
        $sql_str="";
        &_get_db_data($sql_str);
    }
    &_db_close;
}

```

```

    }
}
sub GET_EXP_CHECK_IN
{
    local ($RESERVE_ID,"EXP_CHECK_IN")=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="SELECT RESERVATION.EXP_CHECK_IN FROM RESERVATION";
        $sql_str.=" WHERE (((RESERVATION.RESERVE_ID)=",
        $sql_str.=$RESERVE_ID;
        $sql_str.=")";

        &_get_db_data($sql_str);
        while ($Data->FetchRow())
        {
            $EXP_CHECK_IN =$Data->Data();
        }

        &_db_close;
    }
}
sub SET_RES_STATE_CANCELLED
{
    local ($RESERVE_ID,"RES_STATE")=@_;
    $hoteldb::status = _opendb();
    if ($hoteldb::status == $hotelc::stat_ok)
    {
        $sql_str="update RESERVATION set RES_STATE=";
        $sql_str.=$hotelc::res_cancelled;
        $sql_str.="where RESERVE_ID=";
        $sql_str.=$RESERVE_ID;
        $sql_str.="";
        &_get_db_data($sql_str);
    }

    &_db_close;
}
}

1;

```

Reservok.pl

```

require "cgi_lib.pl";

&ReadParse();

$address = $in{HDADDRESS};
$indate = $in{HDCHECKIN};

```

```

$outdate = ${HDCHECKOUT};
$corp_name = ${HDCORP_NAME};
$name = ${HDNAME};
$phone = ${HDPHONE};
$fax = ${HDFAX};
$sex      = ${HDSEX};
$single = ${TXSINGLE};
$twin = ${TXTWIN};
$suite = ${TXSUITE};
$resvid = ${HDRESVID};

# &createNext();
# &reserror();
&createOK();

sub createOK()
{
    print "HTTP/1.0 200 OK\n";
    print "Content-type: text/html\n";
    print '<HTML><BODY BGCOLOR=WHITE>';

    print '<title>RESERVATION OK!- Update finish.</title>';
    print '<body background=../bg080.jpg bgcolor=#FFFFFF>';
    print '<h3><font color=#FF0080>RESERVATION OK! - Reserved the rooms done</h3>';
    print '<center><table border="1">';
    print '    <tr><td><font size="1">RESERVATION ID</font></td>';
    print '        <td><font size="1"><input type="text";';
    print '            size="14" name="HDRESVID" value="';print $resvid; print '>'</font></td>';
    print '    </tr>';
    print '    <tr><td><font size="1">EXPECTED DATE IN</font></td>';
    print '        <td><font size="1"><input type="text";';
    print '            size="14" name="TXEXPIN" value="';print $indate; print '>'</font></td>';
    print '    </tr>';
    print '    <tr>';
    print '        <td><font size="1">EXPECTED DATE OUT</font></td>';
    print '        <td><font size="1"><input type="text";';
    print '            size="14" name="TXEXPOUT" value="';print $outdate; print '>'</font></td>';
    print '    </tr>';
    print '    <tr>';
    print '        <td><font size="1">Single</font></td>';
    print '        <td colspan="3"><font size="1"><input type="text" size="2" name="TXSINGLE" value="';print $single; print '>'</font></td>';
    print '    </tr>';
    print '    <tr>';
    print '        <td><font size="1">Twin</font></td>';
    print '        <td colspan="3"><font size="1"><input type="text" size="2" name="TXTWIN" value="';print $twin; print '>'</font></td>';
    print '    </tr>';
    print '    <tr>';
    print '        <td><font size="1">Suite</font></td>';
    print '        <td colspan="3"><font size="1"><input type="text" size="2" name="TXSUITE" value="';print $suite; print '>'</font></td>';
    print '    </tr>';
}

```

```

print ' </table>';
print ' <table border="1">';
print ' <tr>';
print ' <td><font size="1">NAME : </font></td>';
print ' <td><font size="1"><input type="text" size="40" name="TXCUST_NAME" value="';print $name; print '></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">ADDRESS : </font></td>';
print ' <td><font size="1"><input type="text" size="40" name="TXCUST_ADDRESS" value="';print $address; print '></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">PHONE </font></td>';
print ' <td><font size="1"><input type="text" size="15" name="TXCUST_TEL" value="';print $phone; print '></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">FAX </font></td>';
print ' <td><font size="1"><input type="text" size="15" name="TXCUST_FAX" value="';print $fax; print '></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">SEX</font></td>';
print ' <td><font size="1"><input type="text" size = "10" name "txsex" value="';print $sex; print '></font></td>';
print ' </tr>';
print ' <tr>';
print ' <td><font size="1">CORP NAME</font></td>';
print ' <td><font size="2"><input type="text" size="20" name="TXCORP_NAME" value="';print $corp_name; print '></font></td>';
print ' </tr>';
print ' </table>';
print '<a href="..">Back to Menu</a>';
print '</center>';
print '</form>';
print '</body>';
print '</html>';
}

```

Ret_resv_id.pl

```

use reservation;
use res_detail;
use hotelc;

require "cgi_lib.pl";
&ReadParse();

%RESV_INF;
$RESV_ID;

$RESV_INF{CUST_NAME}=$in{txname};
$RESV_INF{CUST_ADDR}=$in{txaddress};
$RESV_INF{CUST_TEL}=$in{txphone};

```

```

$RESV_INF{CUST_FAX}=$in{txfax};
$RESV_INF{CUST_SEX}=$in{opsex};
$RESV_INF{EXP_CHECK_IN}=$in{hddatein};
$RESV_INF{EXP_CHECK_OUT}=$in{hddateout};
$RESV_INF{CORP_NAME}=$in{txcorpname};

```

```

#$RESV_INF{CUST_NAME}="TEST MAN";
#$RESV_INF{CUST_ADDR}="TEST ADDRESS";
#$RESV_INF{CUST_TEL}="123456";
#$RESV_INF{CUST_FAX}="098765";
#$RESV_INF{CUST_SEX}="M";
#$RESV_INF{EXP_CHECK_IN}="1997/7/21";
#$RESV_INF{EXP_CHECK_OUT}="1997/8/6";
#$RESV_INF{CORP_NAME}="NEC";

```

```

$ROOM_TYPE[0]=$hotelc::single;
$ROOM_TYPE[1]=$hotelc::twin;
$ROOM_TYPE[2]=$hotelc::suite;

```

```

$NO_OF_ROOMS[0]=$in{hdsingle};
$NO_OF_ROOMS[1]=$in{hdtwin};
$NO_OF_ROOMS[2]=$in{hdsuite};

```

```

#$NO_OF_ROOMS[0]=10;
#$NO_OF_ROOMS[1]=5;
#$NO_OF_ROOMS[2]=1;

```

```

$CORP_DISCOUNT;

```

```

&reservation::RET_DISCOUNT($CORP_NAME,$CORP_DISCOUNT);

```

```

&reservation::CAL_DISC_RATE(@ROOM_TYPE,$CORP_DISCOUNT,$DISCOUNT_RATE);

```

```

&reservation::RET_RESV_ID(%RESV_INF,$RESV_ID);

```

```

if ( $reservation::status==$hotelc::stat_ok )

```

```

{
    &res_detail::UPDATE_RESERVE_DETAIL($RESV_ID,@ROOM_TYPE,@NO_OF_ROOMS);
}

```

```

if ( $reservation::status==$hotelc::stat_ok )

```

```

{ &createNext(); }
else { &reserror(); }

```

```

sub createNext

```

```

{
    print "HTTP/1.0 200 OK\n";
    print "Content-Type: text/html\n\n";
}

```



```

print '</html>';
}

```

Room.pm

```

package rooms;
use hoteldb;
my (%data);
sub NEW
{
    $data{ROOM_NO}=undef;
    $data{ROOM_DESC}=undef;
    $data{ROOM_TYPE}=undef;
    $data{ROOM_RATE}=undef;
    $data{ROOM_STATE}=undef;
    #$data{WA_ROOMS_LIST}=undef;
}

sub CHANGE_STATE
{
    local ($ROOM_NO)=@_;
    &hoteldb::UPDATE_ROOM_STATE($ROOM_NO);
}

sub RET_CHK_OUT_ROOM_LIST
{
    local ($DATE,*WA_ROOMS)=@_;
    &hoteldb::RET_CHK_OUT_ROOM_LIST($DATE,@WA_ROOMS);
}

sub RET_ROOM_STATE_LIST
{
    local (*ROOM_TYPE,$DATE,*ROOM_NO,*ROOM_STATE)=@_;
    &hoteldb::RET_ROOM_STATE_LIST(@ROOM_TYPE,$DATE,@ROOM_NO,@ROOM_STATE);
}

sub RETURN_ROOM_STATE_LIST2
{
    local (*ROOM_TYPE,$DATE,*ROOM_NO,*ROOM_STATE)=@_;
    &hoteldb::RET_ROOM_STATE_LIST(@ROOM_TYPE,$DATE,@ROOM_NO,@ROOM_STATE);
}

sub CHK_VACANCY
{
    local ($ROOM_NO,*VACANT)=@_;
    &hoteldb::CHK_VACANCY($ROOM_NO,$VACANT);
}

sub RET_EMPTY_ROOM_LIST

```

```

{
local ($ROOM_TYPE,*ROOM_LIST)=@_ ;
&hoteldb::RET_EMPTY_ROOM_LIST($ROOM_TYPE,@ROOM_LIST);
}
sub UPDATE_ROOM_STATE
{
local (*WA_ROOM_NO)=@_ ;
&hoteldb::UPDATE_CANCEL_ROOM_STATE(@WA_ROOM_NO);
}
1

```

