**PROJECT REPORT**

ON

**ROOM BOOKING SYSTEM**

By

**Team 4 Members**

**Ms. Dimple Dominic**

**Roll No: 8**

**Ms. Dona Mary Reji**

**Roll No: 9**

**Ms. Josna Mariam Joy**

**Roll No: 11**

**Ms. Sreedevi s**

**Roll No: 16**

Under the guidance of

**Mrs. KavithaS.**

****

**DEPARTMENT OF COMPUTER APPLICATION**

**SAINTGITS COLLEGE OF ENGINEERING, PATHAMUTTOM, KOTTAYAM**

**APRIL 2018**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl.No.** | **Contents** | **Page** |
| 1 | Introduction | 3 |
| 2 | System Environment | 4 |
| 3 | Features of Software: Java | 5 |
| 4 | System Analysis | 9 |
| 5 | System Testing | 18 |
| 6 | Appendix | 24 |
| 7 | Conclusion | 25 |

**INTRODUCTION**

**ABSTRACT**

This is a simple railway reservation program in JAVA where you can book and view the tickets. It also provide the details of the trains. This project deals with reserving tickets for the trains which is heck these days. So in order to make it convenient travel we can book, display our booked seat with our ticket. This makes the travel smooth and convenient.

Some of the features involved in this project are:

Book ticket

Display ticket

Choose your train from a list

**SYSTEM ENVIRONMENT**

**HARDWARE CONFIGURATION**

* Processor - Intel(R)Core(TM) i3-3227U CPU@ 1.90GHz 1.90GHz
* RAM - 4.00GB
* Hard disk - 160GB
* Monitor - Generic PnP Monitor
* Keyboard - 102keys
* Graphics - Direct X9 with 128 memories

**SOFTWARE CONFIGURATION**

* Operating system - Windows 10
* Front end - Java

**FEATURES OF SOFTWARE**

**JAVA**

Java is a powerful general purpose programming languagedevelopedwith the aim to bring portability and a higher level of security.Other than these two main java features, there are many other features of java that makes it such a unique and a popular language.

List of features of JavaProgramming languages-

1. Simple
2. Object Oriented
3. Secured
4. Platform independent
5. Robust
6. Architecture Neutral
7. Portable
8. Dynamic
9. Multithreaded
10. High Performance
11. Interpreted
12. Distributed
13. Simple

Java is said to be a simple language because its syntax is based on the syntax of C++. So it is easy for the programmer to learn Java. In Java, there is concept of automatic garbage collection which frees memory automatically when there is unreferenced object for a long time. There is no use of Pointers rather it provide inbuilt data structures and otherimportant features which makes Java a simple programming language.

1. Object Oriented

Object oriented means that everythingis representedin terms of object which has a state and a behavior. It is a concept or methodology which is used to simulate a real world problems. Java can be easily extended since it is based on the Object model.Object oriented programming makes the software development fast and easy to maintain by providing some concepts. They are:

* Object
* Class
* Inheritance
* Polymorphism
* Abstraction
* Encapsulation

1. Secured

Java is secured because of the following reasons. They are:

* Thereis no concept of explicit pointers in java
* Programs runs inside virtual machine sandbox
* Byte code verifier
* Security manager

It also enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

1. Platform independent

Platform means a hardware/software environment in which a programs runs. In Java, we have only software based platform which consists of:

* Runtime Environment
* API

Java program can run on different platforms such as Windows, Linux, Solaris, Mac OS etc. Java code is compiled by the compiler and is converted to byte code. The byte code which is generated as platform independent code which can run on multiple platform ie.,Write Once and Run Anywhere(WORA).

1. Robust

Simply means strong and powerful. Java puts a lotof emphasis on early checking of possible errors. Java has eliminated certain types of error prone programing constructs found in other languages. Features of java like memory management, automatic garbage collection, type checking mechanism, reliability makes java as robust. Java does not support pointers. It has a runtime exception handling feature to provide programming support for robustness.

1. Architecture Neutral

Today software vendors have developed multiple versions of the same productto runon different platforms. Using Java, developers need towrite only one version that can run on different platform.In some languages like C,Ada etc., the range of the integer varies ondifferent platform. Butin Java, the range of integers are same onevery platform, asitis the behavior of the arithmetic. This features makes Java as architecture neutral.

1. Portable

Java byte code can be carried on any platform. Fixed range of numbers makes Java program portable.

1. Dynamic

Java is designed to adapt to an evolving environment.New classes can be loaded without recompilation. New features can be incorporated transparently when needed. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time

1. Multithreaded

It is a program’s capability to perform several tasks simultaneously. Java program can deals with multiple tasks concurrently by starting multiple threads at once. The main advantage of multithreading is that it shares same memory. Multithreading is necessity in multimedia and network programming. This design feature allows the developers to construct interactive applications that can run smoothly.

1. High Performance

The new JVM uses technology known asJust-In-Time compilation.It compiles Java byte code into native machine code which increases the performances.

1. Interpreted

Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.

1. Distributed

Java is designed for the distributed environment of the Internet.

**SYSTEM ANALYSIS**

System Analysis is the process of gathering and interpreting facts and using the information to recommend improvements to the system. It is detailed study of the various operations performed by a system and their relationship within and outside of the system.

A system can be defined as orderly grouping of independent components inked together according to a plane specified objective. System analysis is a general term that refers to an orderly structure process for identifying and solving problems.

Analysis is the first step towards solving the problem statement. It deals with devising a precise, concise, understandable and correct model of the real system. Analysis helps to understand the requirements and the real world environment to which system will exist. The result of analysis would understand the problems as a preparation for design. After feasibility analysis, the next phase is the study of the current system. The purpose of this phase is to learn how the current system operates. The analyst updates the feasibility estimates and presents the finding as a problem statement for final study of phase reports.

The third phase of system analysis is to define end-user requirements for a new system. The purpose of this phase is to identify what the new and improved information system must be able to do. The product of this phase is the requirement statement. The fourth phase is to select feasible solution from alternative that are evaluated in terms of operational technical and economic feasibility. The analyst will recommend the best solution to the management for approval. A cost benefit analysis determines the expected system development life time, cost for new system and the benefits of the new system.

**SCOPE OF COMPUTERIZATION**

The scope of computerization in any field is increasing speed, accuracy and storage capacity. Here in this project we are implementing an environment for that enables different users can access these services that have a valid Username and Password.

**OBJECTIVE OF COMPUTERIZATION**

The major objective of computerization in any field is to make easier. Our system enables different functions and all users can uses these functions that have a valid Username and Password.

**INITIAL INVESTIGATION**

Initial investigation is the activity that determines whether the user’s requisition is valid and feasible. The first step in initial investigation is the problem definition. It includes the identification of the problem to be solved, for the task to be accomplished and the system goals to be achieved.

**PROBLEM DEFINITION**

**EXISTING SYSTEM**

Even though the system is available but the passengers have to fill all their necessary details using penand papers which involves manual working.It’s the main counter is not pen due to the late arrival or due to some reasonsthen in this case,passengers have to wait andin certain cases they have to wait in line, waiting for their chance to come.After wasting so much time,customers able to get their answers and sometimes they did notpositive response.

**PROPOSED SYSTEM**

Throughthis system,customer do not have to waitin line and they willable to get answers in just a click. They will able tofill their journey details .To book their seats,users will have to search their trainsas per their requirementsand getting correct train.

**MODULE DESCRIPTION**

There are mainly three types of modules:

MODULES ARE:  
  
1.CHECKIN   
CustomerName  
Checkin\_date  
Advance\_amount/not  
 2.BILLING  
 Totalamount=Payment/day\*totaldays  
 Total amount-Advance

3.CHECKOUT  
 CheckoutDate  
 View bill

**DATA FLOW DIAGRAM**

The data flow diagram (DFD) is one of the most important tools used by system analysts. A DFD is also known as “Bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design phase. So it is the starting point of the design phase that functionally decomposes the requirement specifications down to the lowest level of detail.

Data flow diagrams are made up of a number of symbols, which represent system components. Most data flow modeling methods use four kinds of symbols. These symbols are used to represent four kinds of the system components. Processes, data stores, data flows and external entities. Circles in DFD represent processes, Data flow is represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities.

**PURPOSE**

The purpose of the design is to create architecture for the evolving implementation and to establish the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have some reasonably completed model of the behavior of the system. It is important to avoid premature designs, wherein develop designs before analysis reaches closer. It is important to avoid delayed designing where in the organization crashes while trying to complete an unachievable analysis model.

Throughout my project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In my opinion, “efficient design of the data flow and context flow diagrams helps to design the system successfully without much major flaws within the scheduled time”. This is the most complicated part in a project. In the designing process, my project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as a way of expressing system requirements in the graphical form. A data flow diagram also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionally decomposes the requirement specification down to the lowest level. Data Flow Diagrams depicts the information flow, the transformation flow and the transformations that are applied as data move from input to output. Thus DFD describes what data flows rather than how they are processed.

Data Flow Diagram is quite effective, especially when the required design is unclear and the user and analyst need a notational language for communication. It is one of the most important tools used during system analysis. It is used to model the system components such as the system process, the data used by the process, any external entities that interact with the system and information flows in the system.

Data Flow Diagrams are made up of a number of symbols, which represents system components. Data flow modeling method uses four kinds of symbols, which are used to represent four kinds of system components. These are:

* Process
* Data stores
* Data flows
* External entity
* **Process**

Process shows the work of the system. Each process has one or more data inputs and produce one or more data outputs. Processes are represented by rounded rectangles in Data Flow Diagram. Each process has a unique name and number. This name and number appears inside the rectangle that represents the process in a Data Flow Diagram.

* **Data stores**

A data store is a repository of data. Processes can enter data, into a store or retrieve the data from the data store. Each data has a unique name.

* **Data flows**

Data flows show the passage of data in the system and are represented by lines joining system components. An arrow indicates the direction of flow and the line is labeled by name of the data flow.

* **External entity**

External entities are outside the system but they either supply input data into the system or use other systems output. They are entities on which the designer has control. They may be an organizations customer or other bodies with which the system interacts. External entities that supply data into the system are sometimes called source. External entities that use the system data are sometimes called sinks. These are represented by rectangles in the data flow diagram.

Four basic symbols are used to construct data flow diagrams which represent data source, data flows, data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

**Constructing DFD**

Several rules of thumb are used in drawing a DFD. Process should be named and numbered for easy reference. Each name should be a representative of the process. The direction of flow is from top to bottom and from left to right. When a process is exploded into lower-level details, they are numbered. The names of data stores, sources and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

**To construct a Data Flow Diagram, we use**,

* Arrows
* Circles
* Open ended boxes
* Rectangles

**Five rules for constructing a Data Flow Diagram**:

* Arrows should not cross each other.
* Squares, circles and files must bear names.
* Decomposed data flow squares and circles can have same names.
* Choose meaningful names for data flow and process should be numbered for easyreferences.
* The direction of flow is from top to bottom and from left to right

**DFD Symbols**

* A Rectangle defines the source or destination of system data.



* An Arrows identifies flow of data in motion. It is a pipeline through which information flows.
* A circle or bubble represents a process that transforms incoming data flow into outgoing data flows.



* An open rectangle is a data store.



**FEASIBILITY STUDY**

The objective of feasibility study is to test the technical, social and economic feasibility of developing a computer system. This is done by investigating the existing system and generating ideas about a new system. The computer system must be evaluated from a technical viewpoint first, and if technically feasible, their impact on the organization and the staff must be accessed. If a compatible, social and technical system can be devised, then it must be tested for economic feasibility. The 3 important tests for feasibility are :

* Operational feasibility
* Technical feasibility
* Economic feasibility

**Operational Feasibility**

Proposed systems are beneficial only if they can be turned into information systems that will meet the operational requirements of the organization. The test of feasibility asks if the system will work when it is developed and installed. Some of the important questions that are useful to test the operational feasibility of a project are given below:

* Is there sufficient support for the project from the management? From users? If the present system is well liked and use to the extent that people would not be able to see reasons for a change, there may be a resistance.
* Are current methods acceptable to the users? If they are not, users may welcome a change that will bring about a more operational and useful system.
* Have the users been involved in the planning and development of the project, and then the changes of resistance can be possibly reduced.
* Issues that appear to be quite can grow into major problems after implementation.

**Technical Feasibility**

The assessment of technical feasibility must be based on the outline of the system requirements in terms of inputs, outputs, files, programs, procedures and staff. This can be qualified in terms of volumes of data, trends, frequency of updating, etc. Having identified an outline system, the investigator must go on to suggest the type of equipment required, methods of developing the system and methods of running the system.

With regard to the processing facilities, the feasibility study will need to consider the possibility of using a bureau or, if in-house equipment is available, the nature of hardware to be used for data collection, storage, output and processing.

There are number of technical issues, which are generally raised during the feasibility stage of the investigation. They are as follows:

* Does a necessary technology exist to do what is suggested?
* Does the proposed equipment have the capacity to hold the data required to use the new system?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and security?

**Economic Feasibility**

A system that can be developed technically and that will be used if installed must still be profitable for the organization. Financial benefits must equal, or exceed the costs. Justification for any outlay is that it will increase profit and reduce expenditure.

**SYSTEM TESTING**

Software testing is the processes of executing software in a controlled manner, in order to answer the question-Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification.

Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

**Validation :** Are we doing the right job?

**Verification:** Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more through means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, limiting profiles, and test coverage information.

Testing is a set of activity that can be planned in advance and conducted systematically. Testing begins at the module level and work towards the integration of entire computer based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

* Testing is a process of executing a program with the intend of finding an error.
* A good test case is one that has high possibility of finding an undiscovered error.
* A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncovered errors in the software also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program

* Testing For correctness
* For implementation efficiency
* For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult that it may at first appear, especially for large program.

**Test Plan:**

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods the test plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an Independent Test Group (ITG) which is to remove the inherent problemsassociated with letting the builder to testthe thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan .

**TESTING TECHNIQUES**

Following are the testing implemented in this project.

* Unit Testing
* Integration Testing
* User Acceptance Testing
* Data Validation Testing
* Output Testing
* System Testing or Final Testing

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under testing.

**Integration Testing:**

The major concerns of integration testing are developing and incremental strategy that will limit the complexity of the entire actions among components as they are added to the system. Developing a component as they are added to the system, developing and implementation and integration schedules that will make the Modules available when needed, and designing test case that will demonstrate the viability of the evolving system. Though each program works individually, they should also work after linking them together. This is also referred to as interfacing.

Data may be lost across interface and one module can have adverse effect on another. Subroutines are to linking may not do the desired function expected by the main routine. Integration testing is a symmetric technique for constructing programs structure while at the same time conducting tests to uncover errors associated with the interface. In the testing the programs are constructed and tested in small segments.

**System Testing:**

When a system is developed it is hoped that it performs properly. In practice however some errors always occur. The main purpose of testing an information system is to find the errors and correct them. A successful test is the one, which finds and error.

The main objectives of the system testing are:

* To ensure the operation the system in perform as per specification.
* To make sure that system meets user’s requirements during operation.
* To verify that controls incorporate in the system function intend.
* To see that when correct inputs are fed to the system the output are correct.
* To make sure that during operation incorrect input and output will be deleted.

The scope of the system test should include both manual operations and computerized operations. System testing is the comprehensive evolution of the programs, manual procedures, computer operations and controls. System testing is the process of checking if the developed system is working according to original objectives and requirements. All testing needs to be conducted in accordance to the test conditions specified earlier.

**Acceptance Testing:**

An acceptance test has the objective of checking the validity and the reliability of the system. It verifies that the system procedures operate the system specifications and the integrity of vital data is maintained. The system is found to the user-friendly and working effectively.

**Validation Testing:**

As the culmination of integration testing, software is completely assembled as a package. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when software functions in the manner that can be reasonably executed by the user valuators were placed in all the forms and its proper workings on the necessary situations were ensured. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

**Output Testing:**

After performing the validation testing, next step is the output testing of the proposed system, since no system could be useful if it does not produce the desired output in the specified format. The forms were run in order to ensure that the outputs are produced as per the predetermined format.

**SYSTEM IMPLEMENTATION**

Implementation includes all those activities that take place to convert from old system to the new system implementation is the final phase i.e. putting the utility into action. Implementation is the side in the project where theoretical design turned into working system. In this stage the installation of the package in the real environment, to the satisfaction of the intended user and the operation of the system is done. If the implementation of the system is not carefully planned and controlled, it can cause chaos and confusion.

**Implementation Methods**

There are several methods for handling the implementation and consequent conversion from the old system to new computerized system. The most secure method for conversion from the old system is to run the old system and the new system parallel. In this approach, a person may operate in the manual order processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flow. In the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly used method is a direct cut over the existing system to the computerized system. The may be within a week or within a day. There is no parallel activity. However, there is no remedy in case of the problem. This strategy requires careful planning. A working version of the system can also be implemented in one part of the organization and the personal will be piloting the system and changes can be made as and required. But this method is less preferable due to loss of entirety of the system.

Implementation is the process of bringing a developed system into operational use and tur over the user. Implementation activities extend from planning through conversion from old system to the new. At the beginning of the developed phase a preliminary implementation plan is to be created to schedule and manage the many different activities that must be integrated into the plan. The implementation plan is updated throughout the development phase, culminating in a changeover plan for the operational phase.

**APPENDIX**

**DATA DICTIONARY**

Most of the elements undertaken by system analyst in analyzing a problem and designing a system to solve that problem to relate data. Before the design of a new system can begin, all of the data elements used by the system and the relationships between them must be defined and explained. A data element to be the smallest unit of data that is meaningful to the system in which it is used. Often there are structured relationships between data elements. These relationships, called data structures, are characteristics of the data streams and data flow diagrams. Data structures are composed of data elements and other data structures.

When analyzing data flow diagrams, we may consider data flows to be structures in motion and data stores to be data structure at rest. A data flow that could not be further divided would be considered to be data element. Data stores are places in a system where data are stored between processing transactions, they include file and database.

**CONCLUSION**

**SOURCE CODE**

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import java.util.\*;

import java.io.File;

import java.io.\*;

class p1

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

String Option,option1;

do

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("| HOTEL BLACKLE |");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("| Options: |");

System.out.println("| 1. Service Outlets |");

System.out.println("| 2. Estimation About Room Rate |");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.print("Select option: ");

Option = sc.next();

long Amount=0;

long DAmount=0;

int per=10;

switch (Option) {

case "1": System.out.println("SEARCH SERVICE");

System.out.println("ALAPPUZHA");

System.out.println("ERNAKULAM");

System.out.println("KOTTAYAM");

System.out.println("THIRUVANATHAPURAM");

System.out.println("ADOOR ");

break;

case "2":do{

Scanner se = new Scanner(System.in);

System.out.println("OUR AVAILIBILITY ABOUT ROOM");

System.out.println("1. Single Room");

System.out.println("2. Double Room");

System.out.println("3. Family Room");

System.out.println("4. Muliple Room");

System.out.println("5. Booking ");

System.out.println("6. Billing ");

System.out.println("Enter ur choice");

option1=se.next();

switch( option1 ){

case "1":

System.out.println(" HERE WE HAVE SINGLE INDEPENDENT EXCELLENT SPACIOUS ROOM.");

System.out.println(" > It is an Furnished room with Separate wardrobe.");

System.out.println(" > Bed(s) Available. ");

System.out.println(" > And Separate Studytable.");

System.out.println(" > The monthly rent forthis room is Rs:6000.");

System.out.println(" > Food is available which includes Breakfast,Lunch,Evening Snacks,Dinner. ");

break;

case "2":

// Scanner se= new Scanner(System.in);

System.out.println("The room is furnished with bed, dresser and tv. Cable tv and wifi included in rent. ");

break;

case "3":

Scanner nV= new Scanner(System.in);

System.out.println("Designer Family Rooms Some look edgy and modern ");

System.out.println("whileothers seem coordinated and traditional");

System.out.println(" Number of Adults:");

int adult=nV.nextInt();

System.out.println("Number of Children:");

int child=nV.nextInt();

break;

case "4":

Scanner nM= new Scanner(System.in);

System.out.println(" Number of Room:");

int Rom=nM.nextInt();

break;

case "5":Scanner sr= new Scanner(System.in);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("| >>>>>>>>>>>> HOTEL BLACKLE <<<<<<<<<<<<<< |");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Room Type:");

System.out.println("1.Single Room");

System.out.println("2.Double Room");

System.out.println("3.Family Room");

System.out.println("4.Muliple Room");

System.out.println("Enter ur Type");

String type=sr.next();

System.out.println("Name must be entered without space and other characters");

try{

FileWriter fw=new FileWriter("text.txt");

System.out.println("Enter ur Name");

String name=sr.next();

fw.write(name);

System.out.println("Enter ur phnumber");

long phnm=sr.nextLong();

long p=phnm;

fw.write(new Long(p).toString());

String checkindate;

String checkoutdate;

SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy");

Date d1=null;

Date d2=null;

System.out.println("Checkin Date:");

checkindate=sr.next();

fw.write(checkindate);

System.out.println("Checkout Date:");

checkoutdate=sr.next();

fw.write(checkoutdate)

d1 = format.parse(checkindate);

d2 = format.parse(checkoutdate);

//System.out.println(d1);

long diff = d2.getTime() - d1.getTime();

//System.out.println(diff)

long diffDays = diff / (24 \* 60 \* 60 \* 1000);

fw.close();

System.out.println("Enter ur choice");

int choice = Integer.parseInt(type);

switch(choice){

case 1:

Amount=((Amount+6000)\*diffDays);

per=6000/10;

DAmount=((DAmount+6000)\*diffDays)-per;

break;

case 2:

Amount=((Amount+12000)\*diffDays);

per=12000/10;

DAmount=((DAmount+12000)\*diffDays)-per;

break;

case 3:

Amount=((Amount+15000)\*diffDays);

per=15000/10; DAmount=((DAmount+15000)\*diffDays)-per;

break;

case 4: Amount=((Amount+8000)\*diffDays);

per=8000/10;

DAmount=((DAmount+8000)\*diffDays)-per;

break;

default:System.out.println("select a valid option");

break;

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YOUR PAYMENT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("| HOTEL BLACKLE |");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("| NAME : |"+name); System.out.println("| ROOM TYPE : |"+type); System.out.println("| NO OF DAYS : |"+diffDays); System.out.println("| ACTUAL AMOUNT: |"+Amount); System.out.println("|TOTAL/DISCOUNT: |"+DAmount); System.out.println("| \*\*\*\*\*\*\*\*\*\*\*\* |"); System.out.println("..................THANK YOU.........................|");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

BufferedReader br = new BufferedReader(new FileReader("text.txt"));

String line = null;

while ((line = br.readLine()) != null) {

System.out.println(line);

}

System.out.println("Success...");

System.exit(0);

}catch(Exception e){System.out.println(e);

e.printStackTrace();}

case "6":System.exit(0);

default:

System.out.println("Selection Incorrect");

}

}while(option1!=null);

break;

case "3":System.exit(0);

break;

default:

System.out.println("Selection Incorrect");

break;

}

}while(Option!=null);

//sc.close();

}

}

SCREENSHORTS





