**CRUISE MANAGEMENT SYSTEM**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S. NO** | **CONTENT** |
| **1.** | **ABSTRACT** |
| **2.** | **INTRODUCTION** |
| **3.** | **CODE** |
| **4.** | **OUTPUT** |
| **5.** | **CONCLUSION** |
| **6.** | **FUTURE SCOPE** |

**ABSTRACT**

Cruise Management Software is the complete, leading, most advanced system for the cruise ships industry. Increasingly, cruise operators are utilising information and communication technologies (ICTs) to improve service-effectiveness and process efficiency, both on- and off-board. Cruise Management implies managing the travel of passengers through ships. There are companies that arrange travel through ships. People travel through ships for entertainment purposes. Many functions are involved in managing a company that concentrates on Voyage of the passengers from one Location to other.

The software developed through this project must support functions that are related to customer registrations, bookings, tracking the moment of the ships, ship maintenance, ship bunkering, ship inventory, ship procurement and accounting of revenue and expenditure.

Our systems include cruise booking software, guest preference tracking, point of sale hardware and software, Shipboard Management. The proposed Java **project** on travel **management** is an solution to the existing problems regarding **cruise management** in various travel agencies.

**INTRODUCTION**

The Cruise Materials Management System (CMMS) offers the ship operator a complete tool to manage the purchasing and inventory control processes in an efficient, time and cost saving manner.

The program follows the logical purchasing process from time of purchase to the point of consumption and is fully integrated with the Cruise Ship’s Property Management System (SPMS). This program can be used in a shore side location, at a cruise operator’s Head Office (the Head Quarter Module) if they are responsible for the purchasing process for the vessel or fleet of vessels or on board a ship (Shipside Module) if the ship is directly responsible for the purchasing task.

**DESCRIPTION**

Customers for CMMS are the existing Cruise customers who have been using the original Cruise Food & Beverage system but as this program is also a stand-alone product - it can be used independently of the Cruise SPMS - non-Cruise customers are potential customers. In addition to offering full integration with the Cruise SPMS and an on-line interface to the Point of Sale system, one of the key benefits of CMMS is that it gives the user a total overview (historic and real time) of all consumables.

The user can track and check on the status of all food, beverage, uniforms, tools, shop stock, medical items and more, that have been ordered, purchased, delivered, consumed and invoiced throughout the life cycle of that item.

**CODE:**

import java.util.\*;

import java.io.\*;

public class Cruisemanagement

{

public static void main(String[] args)throws IOException {

int i,num;

int choice=0,cho;

float amount,balance=0;

int shipno,customers;

String location,Start,Drop;

String searchshipno;

boolean flag=true;

boolean quit = false;

int snum;

BufferedReader br = new BufferedReader

(new InputStreamReader(System.in));

System.out.println("\n\t\t\tCRUISE MANAGEMENT");

while(flag)

{

System.out.println("\n Menu");

System.out.println("1. Ships data");

System.out.println("2. Customer Booking");

System.out.println("3. Ship Procurement");

System.out.println("4. Ship repairs");

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

System.out.print("Enter your choice: ");

choice= Integer.parseInt(br.readLine());

switch(choice){

case 1:

// Create a file.

FileWriter fout = new FileWriter("test.txt");

//Read data from Keyboard

System.out.println("How many ships? ");

num= Integer.parseInt(br.readLine());

for(i=0;i<num;i++)

{

System.out.println("enter shipno:");

shipno = Integer.parseInt(br.readLine());

System.out.println("Enter ship location:");

location=br.readLine();

System.out.println("Enter ship start point:");

Start=br.readLine();

System.out.println("Enter ship drop point:");

Drop=br.readLine();

customers=0;

System.out.println("\n");//print blankline

// Write to file.

Formatter fmt1 = new Formatter();

fmt1.format("%10d\n%10s \n%10s\n%10s",shipno,location,Start,Drop);

fout.write(fmt1+"\r\012");

}

fout.write("EOF");

fout.close();

System.out.println("File is created");

break;

case 2:

snum=0;

System.out.print("enter Starting and Droping points ");

searchshipno = br.readLine();

//Read data from file

FileReader fin1 = new FileReader("test.txt");

Scanner sc1 = new Scanner(fin1);

while(sc1.hasNextInt())

{

shipno = sc1.nextInt();

location = sc1.next();

Start= sc1.next();

Drop=sc1.next();

if(searchshipno.equals(Start)==true)

{

//Display formatted output

++snum;//increment serial number

Formatter fmt2 = new Formatter();

fmt2.format("%10d.\tshipno.: %10s",snum,shipno);

Formatter fmt3 = new Formatter();

fmt3.format

("\tName: %10s mob no: %10s",location,Drop);

System.out.println(fmt2);

System.out.println(fmt3);

}

}

searchshipno = sc1.next();

if(searchshipno.equals("EOF"))

System.out.println("Search is completed");

else

System.out.println("File format error.");

fin1.close();

break;

case 3:

snum=0;

String Goodstype;

int Goodsweight;

FileWriter fout1 = new FileWriter("test1.txt");

//Read data from Keyboard

System.out.println("How many ships? ");

num= Integer.parseInt(br.readLine());

for(i=0;i<num;i++)

{

System.out.println("enter shipno:");

shipno = Integer.parseInt(br.readLine());

System.out.println("Enter ship location:");

location=br.readLine();

System.out.println("Enter ship start point:");

Start=br.readLine();

System.out.println("Enter ship drop point:");

Drop=br.readLine();

System.out.println("Enter Goods type:");

Goodstype=br.readLine();

System.out.println("Enter Goods weight:");

Goodsweight=Integer.parseInt(br.readLine());

Formatter fmt2 = new Formatter();

fmt2.format("%10d\n%10s \n%10s\n%10s\n%10s\n%10d",shipno,location,Start,Drop,Goodstype,Goodsweight);

fout1.write(fmt2+"\r\012");

}

fout1.write("EOF");

fout1.close();

System.out.println("File is created");

break;

case 4:

String repairtype;

int repaircost;

FileWriter fout2 = new FileWriter("test2.txt");

System.out.println("Repair type");

repairtype = br.readLine();

System.out.println("Enter cost of repair");

repaircost=Integer.parseInt(br.readLine());

Formatter fmt3 = new Formatter();

fmt3.format("%10s\n%10d\n",repairtype,repaircost);

fout2.write(fmt3+"\r\012");

break;

case 5: flag=false;

break;

default:System.out.println("Wrong Choice!!");

}

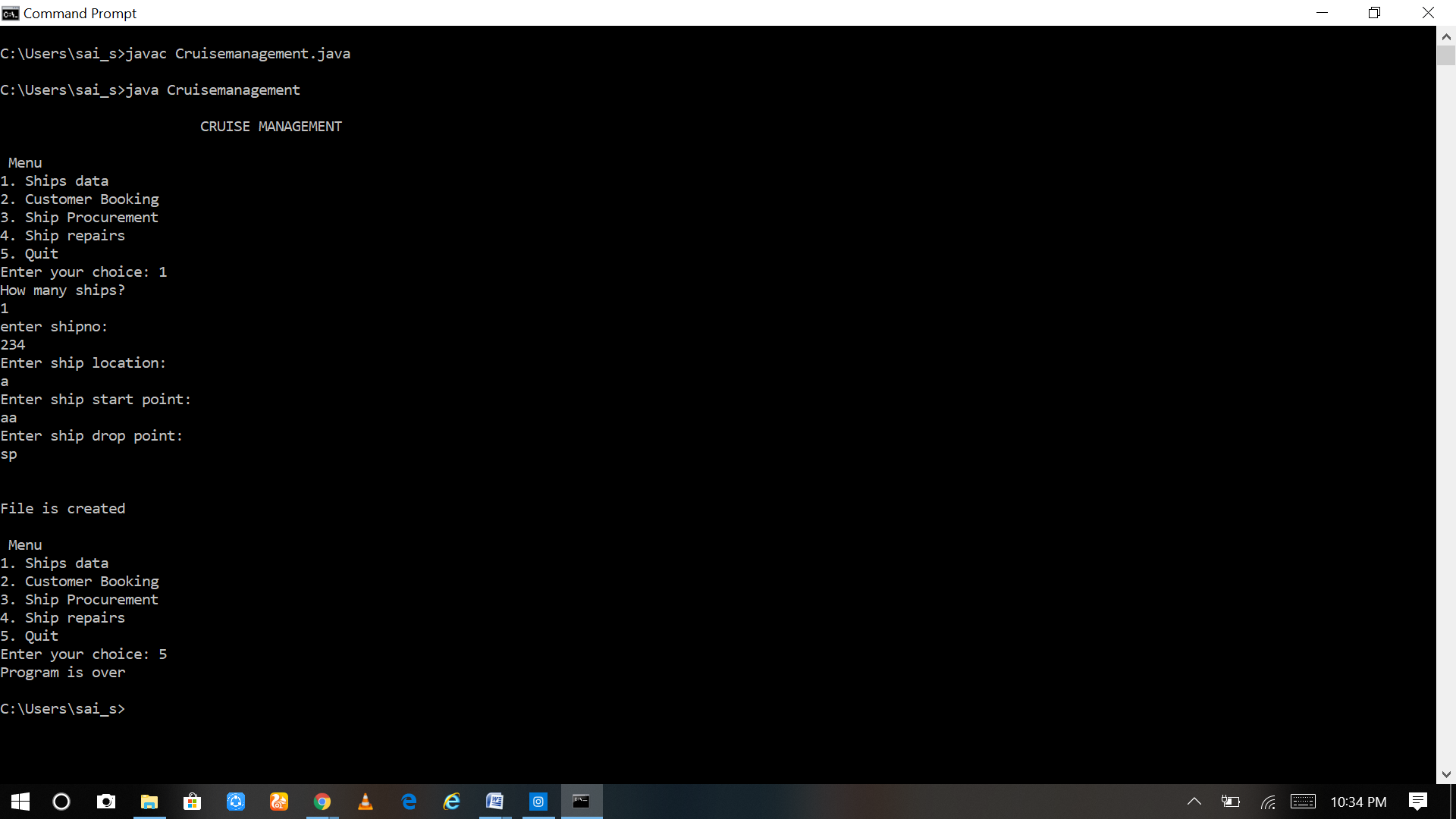
}

System.out.println("Program is over");

}

}

**OUTPUT:**



**Conclusion**

Today’s increasingly digitally enabled ships are bringing more immersive and personalized experiences for guests — and cyber security concerns for IT staff.

**Future Scope**

Some more functions or modules may be added to project. After studying and understanding other ways to implement the code, the inputs and outputs can be improved and implemented.