RAILWAY RESERVATION SYSTEM

21CSC101T – OBJECT ORIENTED DESIGN AND PROGRAMMING

Mini Project Report

I YEAR/ II SEMESTER

Submitted by

NISCHE VERMA[RA2211003010532]

NAVDEEP LAKHLAN[RA2211003010554]

UNDER THE GUIDANCE OF

DR. MURALIDHARAN C.

ASSITANT PROFESSOR

(DEPARTMENT OF COMPUTER TECHNOLOGY)

## **BACHELORS OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE ENGINEERING**

Logo, company name

Description automatically generated

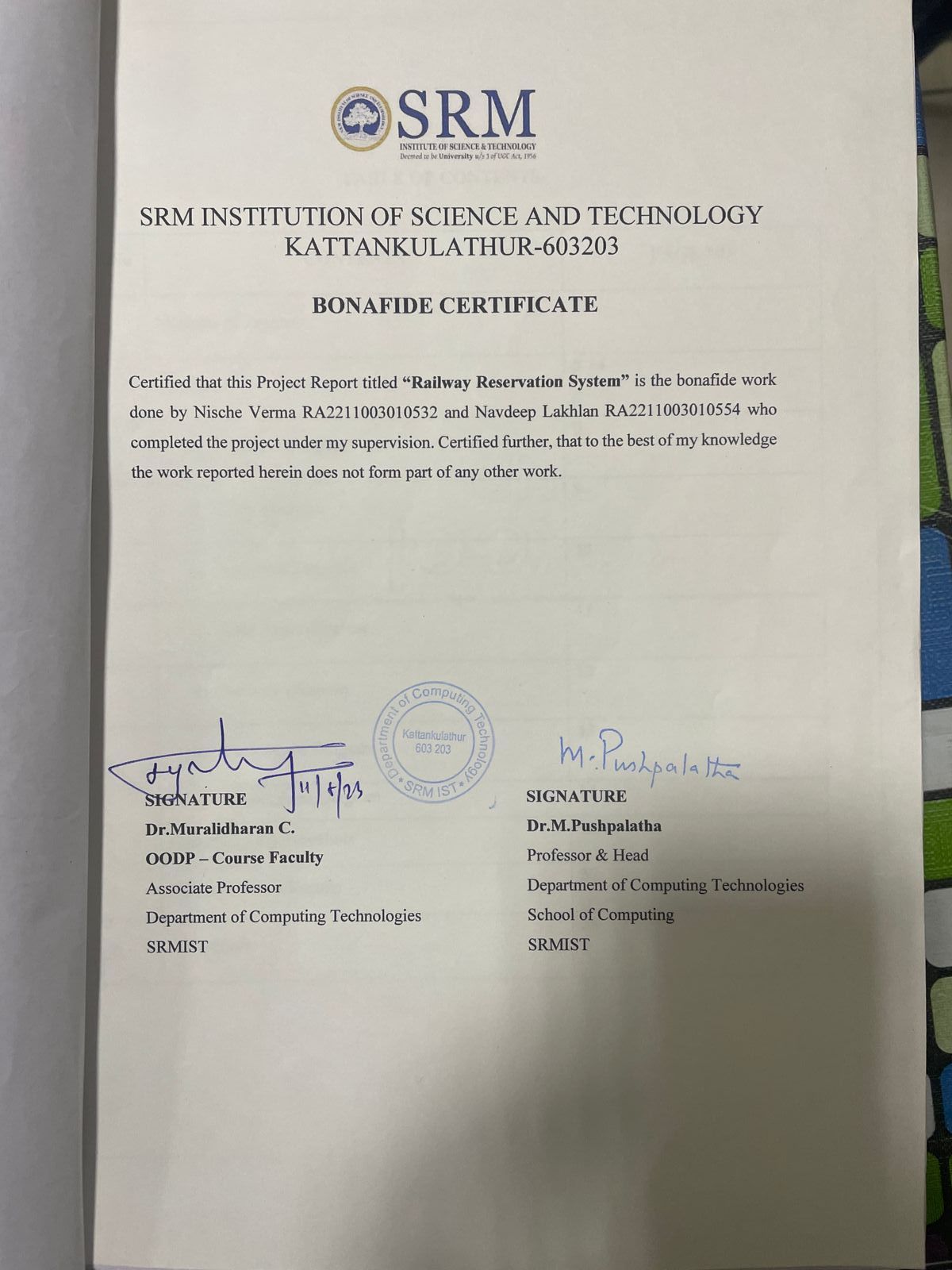
**SCHOOL OF COMPUTING**

# **COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR - 603203**

**MAY 2023**



**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **PAGE NO** |
| 1. | Modules of Project | **7** |
| 2. | Diagrams | **8-14** |
|  | 1. Use case Diagram | **8** |
|  | 1. Class Diagram | **9** |
|  | 1. Sequence Diagram | **10** |
|  | 1. State Chart Diagram | **11** |
|  | 1. Activity Diagram | **12** |
|  | 1. Component Diagram | **13** |
|  | 1. Deployment Diagram | **14** |
| 3. | Code/Output Screenshots | **15-36** |
| 4. | Conclusion and Results | **37** |
| 5. | References | **38** |

**ABSTRACT**

**RAILWAY RESERVATION SYSTEM**

#WHAT IS RA RESERVATION SYSTEM??

|  |
| --- |
| RAILWAY RESERVATION system |
| #1. PASSENGER information |
| #2. OFFICE management |
| #3. Self-management |
| #4. FARE management |
| #5. Notice management |

#WHY DID WE CHOOSE THIS TOPIC AS OUR MINI PROJECT??

The main objective behind choosing this topic as our project is to manage the details of RAILWAY, their PROCEDURE  How RAILWAY BOOKING manages , and other important information like PNR ,fares,  seats ,and etc.

The purpose is to reduce the manual work and to keep a track on everything as well.

MODULE DESCRIPTION

The railway reservation system facilitates the passengers to enquiry about the trains available on the basis of source and destination, booking and cancellation of tickets, enquiry about the status of the booked ticket, etc. The aim is to design and develop a data base maintaining records of different trains, train status and passengers. This project contains introduction to the railways reservation system. It is the computerized system of reserving the seats of train seats in advance. It is mainly used for a long route. Online reservation has made the process for the reservation of seats very much easier than ever before.

This module has been described by 9 diagrams, which are-

1. **Use Case Diagram:** A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.

2. **Class Diagram:** class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

3. **Sequence Diagram:** The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios.

4. **Communication Diagram:** A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

5. **State Chart Diagram:** It define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems.

6. **Activity Diagram:** Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

7. **Package Diagram:** It is used to illustrate how the packages and their elements are organized. It shows the dependencies between distinct packages. It manages UML diagrams by making it easily understandable. It is used for organizing the class and use case diagrams.

8. **Component Diagram:** It portrays the organization of the physical components within the system. It is used for modeling execution details.

9. **Deployment Diagram:** It presents the system's software and its hardware by telling what the existing physical components are and what software components are running on them.

**Use case diagram with explanation.**

A use case diagram can create a broad, high-level view of the relationship between use cases, actors involved, and systems being performed. As you can see from the examples below, use cases are represented by oval shapes, and the lines then show at which point an actor/user participates and interacts with their corresponding use case. You can see where each actor is involved in the entire process.

Diagram

Description automatically generated

**Class diagram with explanation**

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

Diagram

Description automatically generated

**Sequence diagram with explanation**

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

Diagram

Description automatically generated

**State chart diagram with explanation**

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists, and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

Diagram

Description automatically generated

**Activity diagram with explanation**

Activity diagrams in UML display the functionalities of various activities and flow in management processes and software systems. The flow in the activity diagram can be sequential, branched, or concurrent.

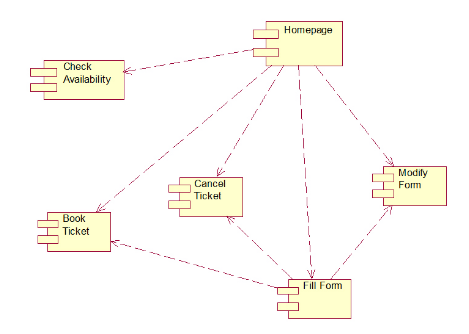
Diagram

Description automatically generated

**Component diagram with explanation**

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.



**Deployment diagram with explanation**

The deployment diagram for student attendance management system is used to represent the system’s physical needs (software and hardware) to ensure it’ll work according to the design given to it. This UML deployment diagram shows the relationships between software and hardware components and the physical distribution of processes.

Diagram

Description automatically generated

**CODING SECTION**

#include <cstdlib>

#include <iostream>

#include <fstream>

#include <string.h>

#include <time.h>

using namespace std;

char f[10] = "f";

char s[10] = "s";

int addr, ad, flag, f1, d, m, i, amt;

float tamt;

class login

{

    public:

    char id[100];

    char pass[100];

    char password[10];

    void getid()

    {

        cout << "\nEnter your id:";

        std::cin >> id;

        std::cout << "Enter the Password: ";

        std::cin >> password;

        std::cout << "\n";

        strcpy(pass,password);

    }

    void displayid()

    {

        std::cout << "\n|" << "User ID:" << "\t" << "|" << "Password\n";

        std::cout << "|" << id << "\t\t" << "|" << pass << "\n\n";

    }

};

class Detail

{

    public:

        int tno;

        char tname[100];

        char bp[100];

        char dest[100];

        int c1,c1fare;

        int c2,c2fare;

        int d,m,y;

        void getDetail()

        {

            std::cout << "\n--Add New details--\n";

            std::cout << "Train no: ";

            std::cin >> tno;

            std::cout << "Train Name: ";

            std::cin >> tname;

            std::cout << "Boarding point: ";

            std::cin >> bp;

            std::cout << "Destination pt: ";

            std::cin >> dest;

            std::cout << "No of seats in first class & fare per ticket: ";

            std::cin >> c1 >> c1fare;

            std::cout << "No of seats in second class & fare per ticket: ";

            std::cin >> c2 >> c2fare;

            std::cout << "Date of travel\n";

            std::cout << "Day: ";

            std::cin >> d;

            std::cout << "Month: ";

            std::cin >> m;

            std::cout << "Year: ";

            std::cin >> y;

        }

        void displaydetail()

        {

            std::cout << "\n|Train No." << "\t" << "|Train Name" << "\t" << "|Boarding pt." << "\t" << "|Destination pt."  <<"\t";

            std::cout << "|F-Class" << "\t"<< "|F-Class Fare" << "\t" << "|S-Class" << "\t" << "|S-Class Fare" <<"\t";

            std::cout << "|Day" << "-" << "Month" << "-" << "Year" << "\n";

            std::cout << "|" << tno << "\t\t" << "|" << tname << "\t\t" << "|" << bp << "\t\t" << "|" << dest <<"\t\t\t";

            std::cout << "|" << c1 << "\t\t" << "|"<< c1fare << "\t\t" << "|" << c2<< "\t\t" << "|" << c2fare <<"\t\t";

            std::cout << "|" << d << "-" << m << "-" << y << "\n";

        }

};

class reser

{

    public:

    int pnr;

    int tno;

    char tname[100];

    char bp[10];

    char dest[100];

    char pname[10][100];

    int age[20];

    char clas[10];

    int nosr;

    int i;

    int d,m,y;

    int con;

    float amc;

    void getresdet()

    {

        cout<<"Enter the details as follows\n";

        cout<<"Train no:";

        cin>>tno;

        cout<<"Train name:";

        cin>>tname;

        cout<<"Boarding point:";

        cin>>bp;

        cout<<"Destination pt:";

        cin>>dest;

        cout<<"No of seats required:";

        cin>>nosr;

        for(i=0;i<nosr;i++)

        {

            cout<<"Passenger name:";

            std::cin >> pname[i];

            cout<<"Passenger age:";

            std::cin>>age[i];

        }

        cout<<"Enter the class f-first class s-second class:";

        std::cin >> clas;

        cout<<"Date of travel:";

        std::cin>>d>>m>>y;

        cout<<"Enter the concession category\n";

        cout<<"1.Military\n2.Senior citizen\n";

        cout<<"3.Children below 5 yrs\n4.None\n";

        std::cin>>con;

        cout<<"����END OF GETTING DETAILS����\n";

    }

    void displayresdet()

    {

        cout<<"Pnr no: "<<pnr;

        cout<<"\nTrain no: "<<tno;

        cout<<"\nTrain name: "<<tname;

        cout<<"\nBoarding point: "<<bp;

        cout<<"\nDestination pt: "<<dest;

        cout<<"\nNo of seats reserved: "<<nosr;

        for(i=0;i<nosr;i++)

        {

            cout<<"\nPassenger name: ";puts(pname[i]);

            cout<<"Passenger age: "<<age[i];

        }

        cout<<"\nYour class: ";puts(clas);

        cout<<"\nDate of reservation: "<<d<<"-"<<m<<"-"<<y;

        cout<<"\nYour concession category: "<<con;

        cout<<"\nYou must pay: "<<amc<<endl;

        cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

    }

};

class canc

{

    public:

    int pnr;

    int tno;

    char tname[100];

    char bp[10];

    char dest[100];

    char pname[10][100];

    int age[20];

    int i;

    char clas[10];

    int nosc;

    int d, m, y;

    float amr;

    void getcancdet()

    {

        std::cout << "Enter the details as follows\n";

        std::cout << "Pnr no:";

        std::cin >> pnr;

        std::cout << "Date of cancellation:";

        std::cin >> d >> m >> y;

        std::cout << "���..END OF GETTING DETAILS���..\n";

    }

    void displaycancdet()

    {

        std::cout << "��������������.\n";

        std::cout << "��������������.\n";

        std::cout << "Pnr no:" << pnr;

        std::cout << "\nTrain no:" << tno;

        std::cout << "\nTrain name:";

        std::cout << tname;

        std::cout << "Boarding point:";

        std::cout << bp;

        std::cout << "Destination pt:";

        std::cout << dest;

        cout << "\nYour class:";

        std::cout << clas;

        cout << "no of seats to be cancelled:" << nosc;

        for (i = 0; i < nosc; i++)

        {

            std::cout << "Passenger name:";

            std::cout << pname[i];

            std::cout << "passenger age:" << age[i];

        }

        std::cout << "\nDate of cancellation:" << d << "-" << m << "-" << y;

        std::cout << "\nYou can collect:" << amr << "rs" << endl;

        std::cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

        std::cout << "���END OF CANCELLATION����.\n";

        std::cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

    }

};

void manage();

void can();

void user();

void database();

void res();

void reserve();

void displaypassdetail();

void cancell();

void enquiry();

int main(void)

{

    int ch;

    std::cout << "----- RAILWAY RESERVATION SYSTEM ----- \n";

    do

    {

        std::cout << "\n MAIN MENU \n";

        std::cout << "1.Admin mode\n2.User mode\n3.Exit \n";

        std::cout << "Enter your choice : ";

        std::cin >> ch;

        switch(ch)

        {

            case 1:

                database();

                break;

            case 2:

                user();

                break;

            default:

                exit(0);

        }

    }while(ch<3);

    return 0;

}

void database()

{

    char password[10];

    char pass[10] = "admin";

    std::cout << "\nEnter the Admin Password: ";

    std::cin >> password;

    Detail a;

    fstream f;

    int ch;

    char c;

    if(strcmp(pass,password)!=0)

    {

        std::cout << "Enter the password correctly \n";

        std::cout << "You are not permitted to login this mode\n";

    }

    if(strcmp(pass,password)==0)

    {

        char c;

        do

        {

            std::cout << "\n --- ADMINISTRATOR MENU --- \n";

            std::cout << "1. Create detail Database \n";

            std::cout << "2. Add details \n";

            std::cout << "3. Display details \n";

            std::cout << "4. User Management \n";

            std::cout << "5. Diplay Passenger details \n";

            std::cout << "6. Return to Main Menu \n";

            std::cout << "Enter your choice : ";

            std::cin >> ch;

            switch(ch)

            {

                case 1:

                    f.open("t.txt",ios::out|ios::binary);

                    do

                    {

                    a.getDetail();

                    f.write((char \*) & a,sizeof(a));

                    cout << "\nDo you want to add one more record?\n";

                    cout << "y-for Yes\nn-for No\n";

                    cin >> c;

                    }while(c=='y');

                    f.close();

                    break;

                case 2:

                    f.open("t.txt",ios::in|ios::out|ios::binary|ios::app);

                    a.getDetail();

                    f.write((char \*) & a,sizeof(a));

                    f.close();

                    break;

                case 3:

                    f.open("t.txt",ios::in|ios::out|ios::binary|ios::app);

                    f.seekg(0);

                    while(f.read((char \*) & a,sizeof(a)))

                    {

                    a.displaydetail();

                    }

                    f.close();

                    break;

                case 4:

                    manage();

                    break;

                case 5:

                    displaypassdetail();

                    break;

            }

        }while(ch<=5);

    }

}

void reserve()

{

    int ch;

    do

    {

        std::cout << "1.Reserve\n2.Return to the main menu\n";

        std::cout << "Enter your choice:";

        std::cin >> ch;

        std::cout << endl;

        switch (ch)

        {

        case 1:

            res();

            break;

        }

    } while (ch == 1);

}

void res()

{

    Detail a;

    reser b;

    fstream f1, f2;

    time\_t t;

    f1.open("t.txt", ios::in | ios::out | ios::binary);

    f2.open("p.txt", ios::in | ios::out | ios::binary | ios::app);

    int ch;

    b.getresdet();

    while (f1.read((char \*)&a, sizeof(a)))

    {

        if (a.tno == b.tno)

        {

            if (strcmp(b.clas, f) == 0)

            {

                if (a.c1 >= b.nosr)

                {

                    amt = a.c1fare;

                    addr = f1.tellg();

                    ad = sizeof(a.c1);

                    f1.seekp(addr - (7 \* ad));

                    a.c1 = a.c1 - b.nosr;

                    f1.write((char \*)&a.c1, sizeof(a.c1));

                    if (b.con == 1)

                    {

                        cout << "Concession category:MILITARY PERSONNEL\n";

                        b.amc = b.nosr \* ((amt \* 50) / 100);

                    }

                    else if (b.con == 2)

                    {

                        cout << "Concession category:SENIOR CITIZEN\n";

                        b.amc = b.nosr \* ((amt \* 60) / 100);

                    }

                    else if (b.con == 3)

                    {

                        cout << "Concession category:CHILDERN BELOW FIVE\n";

                        b.amc = 0.0;

                    }

                    else if (b.con == 4)

                    {

                        cout << "You cannot get any concession\n";

                        b.amc = b.nosr \* amt;

                    }

                    srand((unsigned)time(&t));

                    b.pnr = rand();

                    f2.write((char \*)&b, sizeof(b));

                    b.displayresdet();

                    cout << "\n-----Your ticket is reserved-----\n";

                }

                else

                {

                    cout << "\*\*\*\*\*\*\*\*\*\*Sorry req seats not available\*\*\*\*\*\*\*\*\n";

                }

            }

            else if (strcmp(b.clas, s) == 0)

            {

                if (a.c2 >= b.nosr)

                {

                    amt = a.c2fare;

                    addr = f1.tellg();

                    ad = sizeof(a.c2);

                    f1.seekp(addr - (5 \* ad));

                    a.c2 = a.c2 - b.nosr;

                    f1.write((char \*)&a.c2, sizeof(a.c2));

                    if (b.con == 1)

                    {

                        cout << "Concession category:MILITARY PRESONNEL\n";

                        b.amc = b.nosr \* ((amt \* 50) / 100);

                    }

                    else if (b.con == 2)

                    {

                        cout << "Concession category:SENIOR CITIZEN\n";

                        b.amc = b.nosr \* ((amt \* 60) / 100);

                    }

                    else if (b.con == 3)

                    {

                        cout << "Concession category:CHILDERN BELOW FIVE\n";

                        b.amc = 0.0;

                    }

                    else if (b.con == 4)

                    {

                        cout << "You cannot get any concession\n";

                        b.amc = b.nosr \* amt;

                    }

                    f2.write((char \*)&b, sizeof(b));

                    b.displayresdet();

                    cout << "\n-----Your ticket is reserved-----\n";

                }

                else

                {

                    cout << "\*\*\*\*\*\*\*\*Sorry req no of seats not available\*\*\*\*\*\*\*\n";

                }

            }

        }

        else

        {

            flag = 0;

        }

    }

    if (flag == 0)

    {

        cout << "!\n";

    }

    f1.close();

    f2.close();

}

void displaypassdetail()

{

    std::cout << "\n--------Passenger Details--------\n";

    fstream f;

    reser b;

    f.open("p.txt",ios::in|ios::out|ios::binary);

    f.seekg(0);

    while(f.read((char \*) & b,sizeof(b)))

    {

        b.displayresdet();

    }

    f.close();

}

void enquiry()

{

    fstream f;

    f.open("t.txt",ios::in|ios::out|ios::binary);

    Detail a;

    while(f.read((char \*) & a,sizeof(a)))

    {

            a.displaydetail();

    }

}

void cancell()

{

    Detail a;

    reser b;

    canc c;

    fstream f1, f2, f3;

    f1.open("t.txt",ios::in|ios::out|ios::binary);

    f2.open("p.txt",ios::in|ios::out|ios::binary);

    f3.open("cn.txt",ios::in|ios::out|ios::binary);

    cout<<"\*\*\*\*\*\*\*\*\*\*CANCELLATION MENU\*\*\*\*\*\*\*\*\*\n";

    c.getcancdet();

    while(f2.read((char \*) & b,sizeof(b)))

    {

        if (b.pnr == c.pnr)

        {

            c.tno = b.tno;

            strcpy(c.tname, b.tname);

            strcpy(c.bp, b.bp);

            strcpy(c.dest, b.dest);

            c.nosc = b.nosr;

            for (int j = 0; j < c.nosc; j++)

            {

                strcpy(c.pname[j], b.pname[j]);

                c.age[j] = b.age[j];

            }

            strcpy(c.clas, b.clas);

            if (strcmp(c.clas, f) == 0)

            {

                while (f1.read((char \*)&a, sizeof(a)))

                {

                    if (a.tno == c.tno)

                    {

                        a.c1 = a.c1 + c.nosc;

                        d = a.d;

                        m = a.m;

                        addr = f1.tellg();

                        ad = sizeof(a.c1);

                        f1.seekp(addr - (7 \* ad));

                        f1.write((char \*)&a.c1, sizeof(a.c1));

                        tamt = b.amc;

                        if ((c.d == d) && (c.m == m))

                        {

                            cout << "You are cancelling at the date of departure\n";

                            c.amr = tamt - ((tamt \* 60) / 100);

                        }

                        else if (c.m == m)

                        {

                            cout << "You are cancelling at the month of departure\n";

                            c.amr = tamt - ((tamt \* 50) / 100);

                        }

                        else if (m > c.m)

                        {

                            cout << "You are cancelling one month before the date of departure\n";

                            c.amr = tamt - ((tamt \* 20) / 100);

                        }

                        else

                        {

                            cout << "Cancelling after the departure\n";

                            cout << "Your request cannot be completed\n";

                        }

                        goto h;

                        c.displaycancdet();

                    }

                }

            }

            else if (strcmp(c.clas, s) == 0)

            {

                while (f1.read((char \*)&a, sizeof(a)))

                {

                    if (a.tno == c.tno)

                    {

                        a.c2 = a.c2 + c.nosc;

                        d = a.d;

                        m = a.m;

                        addr = f1.tellg();

                        ad = sizeof(a.c2);

                        f1.seekp(addr - (5 \* ad));

                        f1.write((char \*)&a.c2, sizeof(a.c2));

                        tamt = b.amc;

                        if ((c.d == d) && (c.m == m))

                        {

                            cout << "You are cancelling at the date of departure\n";

                            c.amr = tamt - ((tamt \* 60) / 100);

                        }

                        else if (c.m == m)

                        {

                            cout << "You are cancelling at the month of departure\n";

                            c.amr = tamt - ((tamt \* 50) / 100);

                        }

                        else if (m > c.m)

                        {

                            cout << "You are cancelling one month before the date of departure\n";

                            c.amr = tamt - ((tamt \* 20) / 100);

                        }

                        else

                        {

                            cout << "Cancelling after the departure\n";

                            cout << "Your request cannot be completed\n";

                        }

                        goto h;

                        c.displaycancdet();

                    }

                }

            }

        }

        else

        {

            flag = 0;

        }

    }

    h:

    if(flag==0)

    {

            cout << "Enter the correct pnr no\n";

    }

    f1.close();

    f2.close();

    f3.close();

}

void can()

{

    int ch;

    do

    {

        cout << "�����..CANCELLATION MENU���\n";

        cout << "1.Cancel\n2.Return to the main menu\n";

        cout << "Enter your choice:";

        cin >> ch;

        cout << endl;

        switch (ch)

        {

        case 1:

            cancell();

            break;

        }

    } while (ch == 1);

}

void user()

{

    login a;

    int ch;

    cout<<"\n-----Login User-----\n";

    char password[10];

    fstream f;

    f.open("id.txt",ios::in|ios::out|ios::binary);

    char id[100];

    std::cout << "Enter your ID : ";

    std::cin >> id;

    std::cout << "Enter your Password : ";

    std::cin >> password;

    while(f.read((char \*) & a,sizeof(a)))

    {

        if((strcmp(a.id,id)==0)&&(strcmp(a.pass,password)==0))

        {

            do

            {

                std::cout<<"\n1.Reserve\n2.Cancel\n3.Enquiry\n4.Return to the main menu\n";

                cout<<"Enter your choice:";

                cin>>ch;

                switch(ch)

                {

                    case 1:

                        reserve();

                        break;

                    case 2:

                        cancell();

                        break;

                    case 3:

                        enquiry();

                        break;

                }

            }while(ch<=3);

        }

        else

        {

            d=1;

        }

    }

    if(d==1)

    {

        cout<<"Enter your user id and password correctly\n";

    }

}

void manage()

{

    int ch;

    fstream f;

    char c;

    login a;

    std::cout<<"\n---WELCOME TO THE USER MANAGEMENT MENU---\n";

    do

    {

        std::cout << "1. Create id database\n";

        std::cout << "2. Add details\n";

        std::cout << "3. Display details\n";

        std::cout << "4. Return to the main menu\n";

        std::cout << "Enter your choice: ";

        std::cin>>ch;

        switch(ch)

        {

            case 1:

                f.open("id.txt",ios::out|ios::binary);

                do

                {

                    a.getid();

                    f.write((char \*) & a,sizeof(a));

                    std::cout << "Do you want to add one more record\n";

                    std::cout << "y-Yes\nn-No\n";

                    std::cin>>c;

                }while(c == 'y');

                f.close();

                break;

            case 2:

                f.open("id.txt",ios::in|ios::out|ios::binary|ios::app);

                a.getid();

                f.write((char \*) & a,sizeof(a));

                f.close();

                break;

            case 3:

                f.open("id.txt",ios::in|ios::out|ios::binary);

                f.seekg(0);

                while(f.read((char \*) & a,sizeof(a)))

                {

                    a.displayid();

                }

                f.close();

                break;

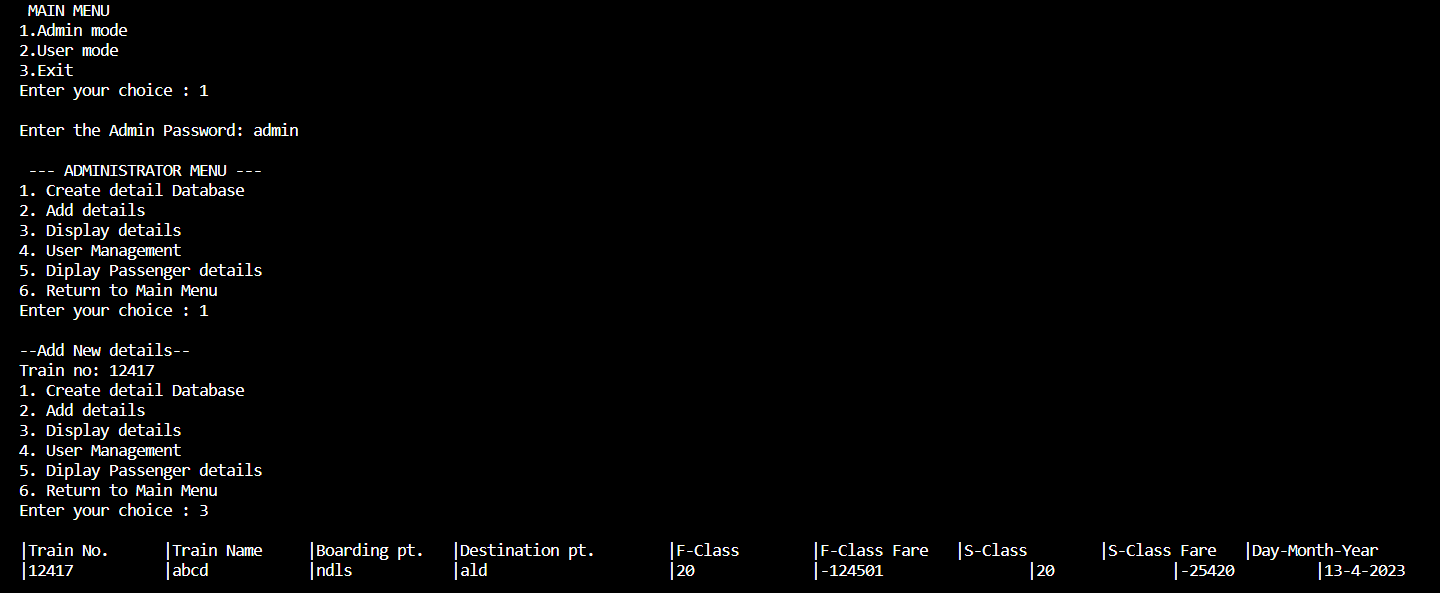
            }

        }while(ch<=3);

}

**OUTPUT SCREEN**

**ADDING OF TRAIN DETAILS**

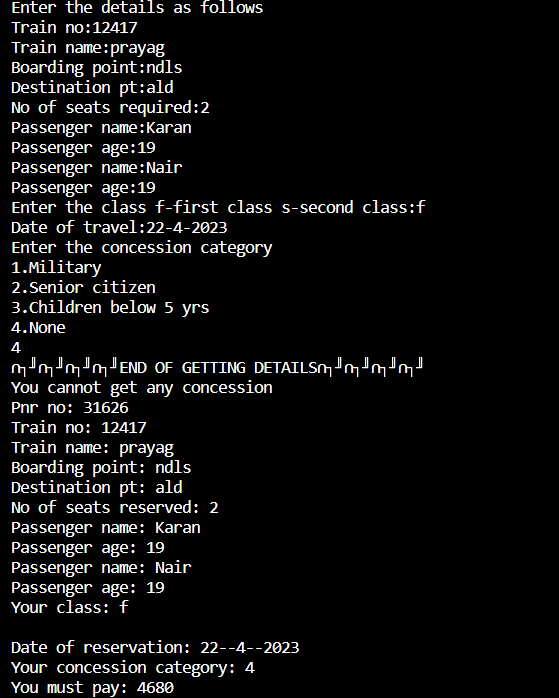
****

**CREATING THE USER\_ID**

**Text

Description automatically generated**

**RESERVATION**

****

**PASSENGER DETAILS**

**Text

Description automatically generated**

**REMAININF OF SEATS AND TRAIN INFORMATION**

**Graphical user interface

Description automatically generated with low confidence**

**CANCELLATION PROCESS**

**A city at night

Description automatically generated with low confidence**

**Conclusion**

We have therefore understood the module “RAILWAY RESERVATION SYSTEM” by seeing and understanding various UML diagrams. The RAILWAY RESERVATION System allows us to see booking report of train and other work done in the railway. We have observed many aspects for railway reservation System to work and conceptualized how different factors impact the working of our module.

**References**

* • Javapoint- UML Diagrams - Javatpoint
* • Wikipedia- Unified Modeling Language - Wikipedia
* • Wikipedia- Attendance Management System - Wikipedia
* • Visual Paradigm- What is Activity Diagram? (visual-paradigm.com)