**Pseudocode: Bus Reservation System**

//Pseudocode for main

Start:

In this we have functions

function of welcomescreen():

print welcome to bus reservation management system

function of mainMenu():

declare int choice;

print main menu

print 1.admin menu

print 2.usermenu

print 3.Exit

print Enter your choice

read choice

By using switch case select the choice

If choice is set to 1

Calls function adminLogin()

declare string name, string password ;

print"ADMIN LOGIN";

read adminname and password;

check if(admin name and password are correct)

calls adminmenu();

else

invalid adminname and password

calls function usermenu():

else if choice is set to 2

print usermenu;

print 1.Book Ticket

print 2.view Ticket

print 3.Cancel Ticket

print 4.Exit

print Enter your choice

read choice

By using switch case enter your choice

function adminMenu():

declare bus b, ticket t;

declare int choice;

print "ADMIN PORTAL"

print 1.Add Bus\n2.view buses details\n3.Edit bus\n4.delete bus\n5.View booking menus\n6.back;

print enter your choice

read choice

by using switch case enter your choice

function viewbookingsmenu():

declare ticket t;

declare int choice;

print "VIEW BOOKINGS"

print 1.By PNR\n2.By name\n3.by bus\n4.by source\n5.by destination\n6.all bookings\n7.back

print enter your choice;

read choice

by using switch case enter your choice

choice is not present choose valid option

//Pseudocode: bus headerfile

class Bus

private:

declare int maxseats, bookedseats, Double busfare, char busno, source, destination, sourcetime, destination time;

public:

bus()//constructor

busno,maxseats,bookedseats,busfare,source,destination,sourcetime,destiation time;

function name addbus();

function name viewbus details();

function name edit bus();

p

function name delete bus();

declare getter and setter functions for all attributes;

Pseudocode: bus definition file:

class Bus :: function addbus()

file open

print "ADD BUS"

print all the details of bus like bus number, source, destination etc..

Read all the above details

file close

print "BUS ADDD SUCCESSFULLY";

class Bus :: Showbusdetails()

print all the bus details like busno, source, destination, fare, time;

class Bus :: function showallbus()

open the file fstream

check file (if file is not found)

else

print "BUSES"

open the busfilestream

read the data

while file found

display busdetails

close the file;

class Bus :: function viewbusdetails()

declare bno;

int ch;

open the file

print "VIEW BUS INFO"

check if(file is opened or not)

if(can't open file)

else

open the file and read the characters

check if (busno==0)

show bus details

else

bus not found

close the file

class Bus :: function edit bus()

declare bno, int ch;

open the file fstream bus and temp

print"EDIT BUS"

print bus number

read busnumber

check(if bustream file opened or not)

if(can't open file)

else

open the file

if(compare the getter function busno and bno)

print"EDIT BUS"

display show busdetails();

declare

char source, destination, time;

double fare;

print and read all the above details;

by using set method write the data into tempfilestream

else

open the two files

if(ch==1)

print bus updated successfully

else

bus not found

close the read and write files

remove buses data and rename the temp and buses data

class Bus :: function deletebus()

declare char bno, int ch;

open the file bus and temp;

print"DELETE BUS"

print busnumber;

read bus number;

open to file busstream

check the file is opened or not

if(file not opened)

else

temp file is opened and bus file will read

compare the getter busno and busno

if(!=0)

open the tempfile to write the data

else

ch=1;

open the busfile to read

if(ch==0)

print bus not found

else

bus deleted

close the bus and temp files

remove the bus data and rename the temp and bus data

end;

class Ticket

private:

declare char name, PNR no, date, bus;

public:

display the functions:

void generate ticket():

void display ticket():

void book ticket():

void cancel ticket():

void edit ticket():

void show ticketby PNR():

void show ticketby name():

void show ticket by bus():

void show ticket by source():

void show ticket by destination():

declare all getter and setter functions for private;

//definition file for ticket booking

in the main function

//display ticket details

void ticket :: display\_ticket();

{

print book tickets

if (busfilestream.fail())

print can't open file

else

{

call set source, set destination

print available buses

print please select bus from available buses

read from user

}

if user input is equal to zero

{

print no buses found

else

{

print enter bus number 10 and get bus number 10

}

if bus number is equal to source and destination getbookseats

if getbookedseats is greater then equal to 32

{

print seats not available

else

{

ask user to fill details for booking ticket

print heading booking details

}

}

// display ticket

display();

print ticket booked successfully

//function for cancel ticket

void ticket :: cancel\_ticket();

{

print enter PNR number

if ticketfilestream is fail

print can not open the file

else

close the ticketstream

if (check == 0)

print ticket not found

else

print ticket cancelled

}

//void ticket :: edit\_ticket

print enter PNR number

if ticketfilestream is fail

{

print can not open file

else

{

while (!ticketfilestream.eof())

if PNR number is equal to zero

print edit ticket enter passenger name

print updated successfully

}

else

print ticket not found

}

//function for show ticket by PNR

void ticket :: showticketsbypnr();

printheading show bookings by phone number

print enter PNR number

cin.getline PNR number must be 10 digits

if ticketfilestream is fail

{

print can not open file

else

{

read the file from ticketfilestream.read((char\*)this, size of(\*this));

print no bookings....

}

}

//show ticket by name

void ticket :: showticketbyname()

{

class Ticket

private:

declare char name, PNR no, date, bus;

public:

display the functions:

void generate ticket():

void display ticket():

void book ticket():

void cancel ticket():

void edit ticket():

void show ticketby PNR():

void show ticketby name():

void show ticket by bus():

void show ticket by source():

void show ticket by destination():

declare all getter and setter functions for private;

//definition file for ticket booking

in the main function

//display ticket details

void ticket::display\_ticket();

{

print book tickets

if (busfilestram.fail())

print can't open file

else

{

call set source, set destination

print available buses

print please select bus from available buses

read from user

}

if user input is equal to zero

{

print no buses found

else

{

print enter bus number 10 and get bus number 10

}

if bus number is equal to source and destination getbookseats

if getbookedseats is greater then equal to 32

{

print seats not available

else

{

ask user to fill details for booking ticket

printheading booking details

}

}

// display ticket

display();

print ticket booked successfully

//function for cancel ticket

void ticket :: cancel\_ticket();

{

print enter PNR number

if ticketfilestream is fail

print can not open the file

else

close the ticketstream

if (check == 0)

print ticket not found

else

print ticket cancelled

}

//void ticket :: edit\_ticket

print enter PNR number

if ticketfilestream is fail

{

print can not open file

else

{

while (!ticketfilestream.eof())

if PNR number is equal to zero

print edit ticket enter passenger name

print updated successfully

}

else

print ticket not found

}

//function for show ticket by PNR

void ticket :: showticketsbypnr();

printheading show bookings by phone number

print enter PNR number

cin.getline PNR number must be 10 digits

if ticketfilestream is fail

print cannot open file

else

read the file from ticketfilestream.read((char\*)this, sizeof(\*this));

print no bookings....

}

}

//show ticket by name

void ticket::showticketbyname()

{

printheading show booking by name

print enter passenger name

ticketfilestream.open("ticket.date", ios::on | ios::app | ios::binary);

if tickekfilestream is fail

print this file can not open

else

get name is equal to zero

print no bookings

}

//function for show ticket by bus

void ticket :: showticketbybus()

printheading showticketbybus

ticketfilestram.open(""ticket.date", ios::in | ios::app | ios::binary);

if ticketfilestream is fail

{

print can not open file

}

else

{

print no bookings...

}

//show ticket by source

void ticket :: showticketbysource()

printheading showticketbysource

print enter source

ticketfilestram.open(""ticket.date", ios::in | ios::app | ios::binary);

if ticketfilestream is fail

{

print file can not open

}

else

{

if bus get source is equal to zero

print no booking

}

//show ticket by destination

void ticket :: showticketbydestination()

printheading showticketbydestination

print enter destination

ticketfilestram.open(""ticket.date", ios::in | ios::app | ios::binary);

if ticketfilestream is fail

{

print file can not open

}

else

{

if bus get destination is equal to zero

print no booking

}

//show all booking

void ticket :: showalltickets()

printheading booking

ticketfilestram.open(""ticket.date ",ios::in | ios::app | ios::binary);

if ticketfilestream is fail

{

print file can not open

}

else

{

ticketFileStream.read((char \*)this, sizeof(\*this));

displayticket

ticketfilestream closed

}