MASTER OF COMPUTER APPLICATION

MCSL - 036 (Section-2 : Software Engineering) – Lab

Name : **Gowtham R**

Enrolment Number : **187012107**

Year / Semester : **II / I**

Study Centre Code : **2548**

Study Centre : **Kongu Engineering College**

DIRECTORATE OF DISTANCE EDUCATION

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

New Delhi

Certified that this is the bonafide record work done by **Mr. R. Gowtham** with Enrolment Number: **187012107** of MCA (Master of Computer Applications) – **III Semester** in the Programming Lab – **MCSL-036** in the directorate of Distance Education, Indira Gandhi National Open University, New Delhi, during the Calendar year 2019 – 2020

Date:

Submitted for MCA Degree course Practical Examination held on **08.01.2020** at the center **G.R.D. College of Science, Coimbatore.**

Date :

Examiners

1. Name :
   Signature :
2. Name :
   Signature :

Software Engineering – LAB

Index

SOFTWARE ENGINEERING - LAB

Session – 1

Scope of the Railway Reservation System

Reservation Clerk is a person to create and Cancel Reservation by entering Login Password. Manager is a person who updates the Train Information by entering his own Password. The system can handle only reservation and train details efficiently and it doesn't contain other details about the railway administration. The main purpose of this system is

- Creating reservation
- Cancel reservation
- View reservation status
- View train schedule
- Generating reports
- Update train schedule
- Update reservation details

The Seats of Reservation cannot be more than the seats of Train at that date. This is a constraint that has to be followed by the Clerk when he creates the Reservation. For that purpose, he wants to check the seats remaining present in the Train.

The scope of this system in creating Reservation is that, from any Railway Station we can Create Reservation, which is updated automatically in all the stations. Hence, there is no confusion to the Reservation Clerk in all the stations to create the Reservation. This can be possible by maintaining Global Database. Clerks present at different stations can access the global database and the clerks can easily understand the remaining reservation seats. It provides the ability to create reservation from different places for a train.

The system is so secured and clerk and manager utilize it. Nobody can able to access the system without his or her permission because of providing login facility to the system. The password is in the form of cipher text by using cryptography technology, so it cannot be hacked by any person. The global database can pass through network in order to utilized by managers or clerks at different places. So, we want to provide network security because of the data not hacked by the other persons when it is going through network cables. This network security is provided by the cryptography technology.

Session – 2

This session is to estimate the size, effort, schedule, and cost of the railway reservation project.

The four basic steps in Railway Reservation project estimation are:

1. Estimate the size of the development product. This generally ends up in either Lines of Code (LOC) or Function Points (FP), but there are other possible units of measure. A discussion of the pros & cons of each is discussed in some of the material referenced at the end of this report.
2. Estimate the effort in person-months or person-hours.
3. Estimate the schedule in calendar months.
4. Estimate the project cost in dollars (or local currency)

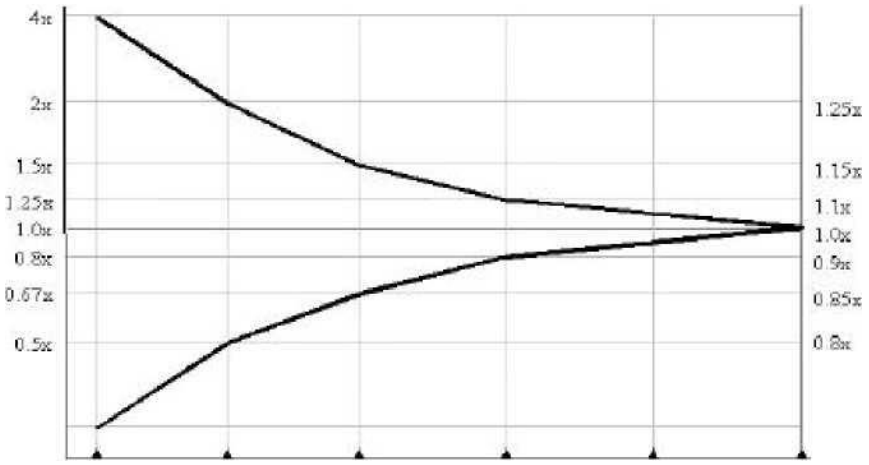
The railway reservation project requires 75KBLOC. On the basis of this size of the project we have to estimate the remaining estimations. We have the formula to estimate the schedule of the project.

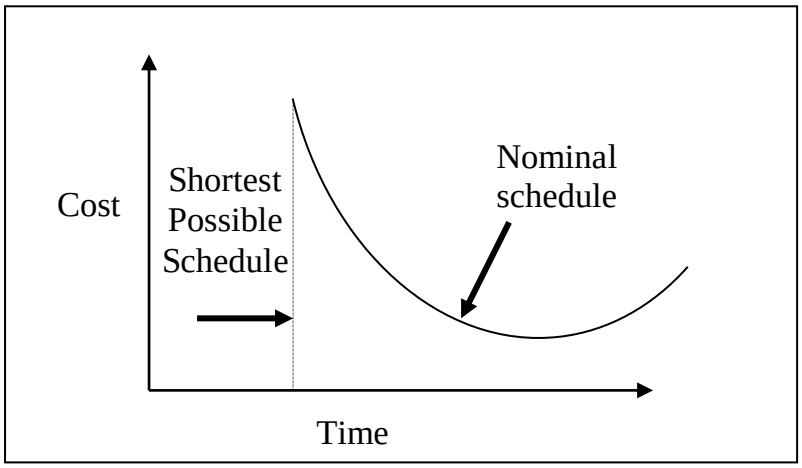
$$\text{Schedule in months} = 3.0 * (\text{effort-months})^{1/3}$$

Opinions vary as to whether 2.0 or 2.5 or even 4.0 should be used in place of the 3.0 value - only by trying it out will you see what works for you.

Of course, you must also keep in mind other important factors that affect the accuracy of your estimates, such as:

- The accuracy of all the estimate's input data (the old adage, "Garbage in, Garbage out", holds true)
- The accuracy of any estimate calculations (e.g., converting between Function Points and LOC has a certain margin of error)
- How closely the historical data or industry data used to calibrate the model matches the project you are estimating
- The predictability of your organization's software development process, and whether or not the actual project was carefully planned; monitored and controlled, and no major surprises occurred that caused unexpected delays.





When they analyze the results, most people are startled by the consequences of different tradeoffs. For example, the following tables provide 3 different estimate options for a 75 KLOC project:

The difference between the nominal schedule and the shortest schedule for the project is only a little over two months, but to achieve the shortest schedule the peak staff has to increase by almost 10 people and the cost increases by over $870,000! These results should cause someone to ask if a 2-month decrease in the schedule is worth the cost, and if 10 additional people can be found in time to help achieve it. For some projects, a schedule decrease may be required at any cost; for others, it won't be.

Nominal Plan

Management Metric	**Planning Value**
Effort (Staff months)	40
Schedule (Calendar months)	12.4
Cost	$605,868
Peak staff (People)	4.8
Average staff (people)	3.2

Shortest Schedule – Plan

Management Metric	**Planning Value**
Effort (Staff months)	97
Schedule (Calendar months)	10
Cost	$1,479,170
Peak staff (People)	14.6
Average staff (people)	9.8

Least Cost – Plan

Management Metric	**Planning Value**
Effort (Staff months)	14
Schedule (Calendar months)	16.2
Cost	$212,131
Peak staff (People)	1.3
Average staff (people)	0.9

Session – 3

Software Requirement Specification for Railway Reservation System

The SRS for Railway Reservation System is given as follows:

Introduction

- Purpose
- Scope
- Definition
- Product and its function
- Benefits and Goals

Overall Description

- Product Description
- Product Functioning
- Functions of Project
- Users of Project

Specific Requirements

- Interface Requirements
- User Requirements
- Hardware Requirements
- Software Requirements
- Logical Database Requirements

Basic Processing Action of the System Appendices

- Input/Output Formats
- Instruction for Security
- Data Model
- Functional Model

Introduction

Purpose:

The purpose of Railway Reservation System is to create Reservation, Cancel Reservation, Viewing Train Information, Viewing Reservation Details, Updating Train Information, Updating Reservation Details and Generate Reports.

Scope:

Reservation Clerk is a person to create and Cancel Reservation by entering Login Password. Manager is a person who updates the Train Information by entering his own Password. The system can handle only reservation and train details efficiently and it doesn't contain other details about the railway administration. The main purpose of this system is

- Creating reservation
- Cancel reservation
- View reservation status
- View train schedule
- Generating reports
- Update train schedule
- Update reservation details

The Seats of Reservation cannot be more than the seats of Train at that date. This is a constraint that has to be followed by the Clerk when he creates the Reservation. For that purpose he wants to check the seats remaining present in the Train.

The scope of this system in creating Reservation is that, from any Railway Station we can Create Reservation, which is updated automatically in all the stations. Hence, there is no confusion to the Reservation Clerk in all the stations to create the Reservation. This can be possible by maintaining Global Database. Clerks present at different stations can access the global database and the clerks can easily understand the remaining reservation seats. It provides the ability to create reservation from different places for a train.

The system is so secured and clerk and manager utilize it. Nobody can able to access the system without his or her permission because of providing login facility to the system. The password is in the form of cipher text by using cryptography technology, so it cannot be hacked by any person. The global database can passes through network in order$to utilized!by managers!or clerks at different places. So, we want to provide network security because of the data not hacked by the other persons when it is going through network cables. This network security is pzovided by the cryptograrhy technology.

Definition:

This is a project, which is used to Create and Cancel Reservation and to Update the Train and Reservation details.

Product & its functions:

The Product of this project is Railway Reservation System, which is to create Reservation, Cancel Reservation, Viewing Train Information, Viewing Reservation Details, Updating Train Information, Updating Reservation Details and Generate Reports.

Benefits and Goals:

The Benefit of this project is to reduce the work of Reservation Clerk and it is easy to check the remaining seats present in the Train and easily view the Reservation Status of the Train and tells to the passenger.

The Goals of this project is that, from any Railway Station we can Create Reservation, which is updated automatically in all the stations. Hence, there is no confusion to the Reservation Clerk in all the stations to create the Reservation.

Overall Description

Product Description:

The Reservation Clerk gets the details from the passenger and he checks whether the weats are remaining in order vo grecte`the Reservation. If the seats are available, then he allocate the seat to the passenger by giving the`Reservation Slip to the passenger and he Update the Reservation details. If weats`are not`availqble, hg places the Rewervation in 'Waiting List'.

The Resmrvation Clerk gets the details from the passenger in order to cancel the Reservation and Canwels Reserva~ion. He also$Update the!Reserva|ion details after canceling the Reserva|ion and he creates the Reservction gor the Passenger who are placed in the Waiting List.

The reservation clerk tell the reservation status to passengers who are in waiting list.

The manager updates the train information and he generates the report of the train, reservation details.

Product Functioning:

- The reservation clerk takes the detail from the passenger.
- The reservation clerk checks whether the seats are reaming or not.
- If seats are available, reservation clerk create the reservation and updates the reservation details.
- If seats are not available he place the reservation in waiting list.
- Reservation clerk cancels the reservation and update the reservation details.
- The manager updates the train information and generates the report.

Functions of Project:

There are seven functionalities provided by the Railway Reservation System.

1. Create Reservations: A passenger should be able to reserve seats in the train. A reservation form is filled by the passenger and given to the clerk, who then checks for the availability of seats for the specified date of journey. If seats are available them the entries are mode in the system regarding the train name, train number, date of journey, boarding station, destination, person name, sex and total fare. Passenger is asked to pay the required fare and the tickets are printed. It the seats are not available then the passenger is informed.

2. Cancel Reservation: A passenger wishing to cancel a reservation is required to fill a form. The passenger then submits the form and the ticket to the clerk. The clerk then deletes the entries in the system and changes the reservation status of that train. The clerk crosses the ticket by hand to mark as cancelled.

3. Update Train Info: Only the administrator or manager enters any changes related to the train information like change in the train name, train number, train route etc. in the system.

4. Generate Report: Provision for generation of different reports should be given in the system. The system should be able to generate reservation chart, monthly train report etc.

5. Verify login: For security reasons all the users of the system are given a user id and a password. Only if the id and password are correct is the user allowed entry to the system and select from the options available in the system.

6. View Reservation Status: All the users should be able to see the reservation status of the train online. The users needs to enter the train number and the pin number printed on his ticket so that the system can display his current reservation status like confirmed, RAC or Wait – Listed.

7. View Train Schedule: Provision should be given to see information related to the train schedules for the entire train network. The user should be able to see the train name, train number, boarding and destination stations, duration of journey etc.

Users of Project:

Clerk: This person uses this system to create reservation, cancel reservation, view reservation status, update reservation details, view train schedule.

Manger: This person uses this system to update train information and to genercte reports.

Specific Requirements

Interface Requirements:

The interface requirements of Railway Reser~ation System include

I

- Easy to Navigate.
- Less Graphics
- Display Error Messages ane Relevant Dialogue Boxes
- Providing high security such that not to be modified by irrelevant users.
- It must provide anl options such that(to reduce the input cctions by usevs.

User Requirements:

After a brief study of requirements of clients the requirmments of this system is given as follows:

- Login information
- Reservation details
- Train details

Hardware Requirements:

The hardware requirements of this system is given by

- ✓ 128 MB of RAM
- ✓ 40 GB of HDD
- ✓ Printer
- ✓ LAN Cable

Software Requirements:

The software requirements of this are as follows:

- ✓ Windows 2000 OS
- ✓ SQL

Logical Database Requirements:

The following information is to be stored in the database

- ✓ Passenger details
- ✓ Reservation details
- ✓ Train details
- ✓ Login details

Basic Processing Actions of the System

The basic actions that are performed by the system is given as follows:

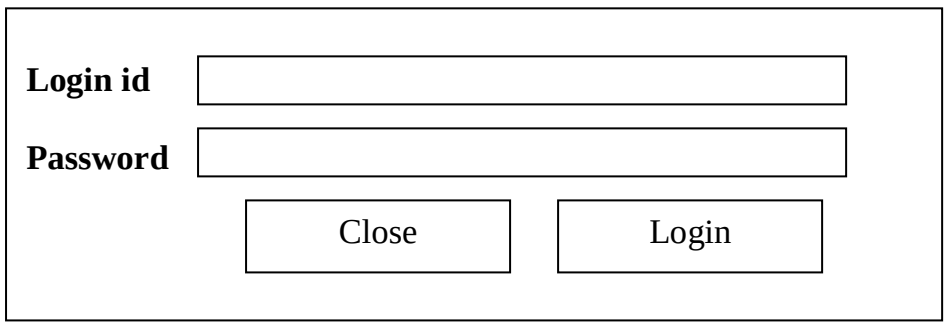
- ➢ Verification of user
- ➢ Maintain Reservation details
- ➢ Maintain Train details

Appendices

Input/Output formats:

The input format for the system is given as follows:

- ➢ Login Screen



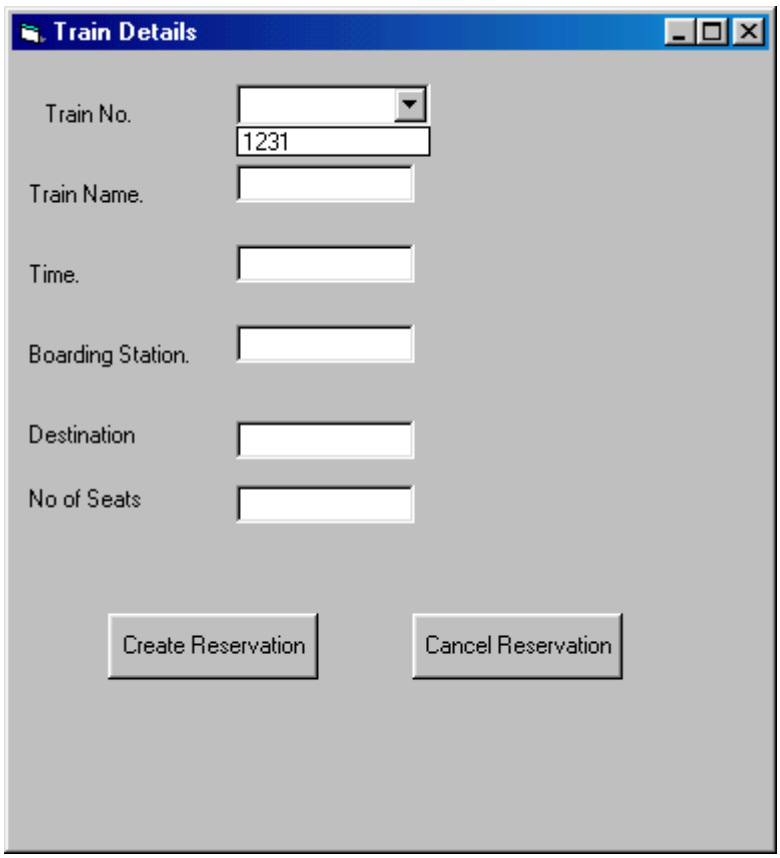
Create Reservation

Train No		Date of Journey			
From		To			
Sl. No	Name of the Persons		Age	M/F	Requirement

Output format for the system contains the following screen

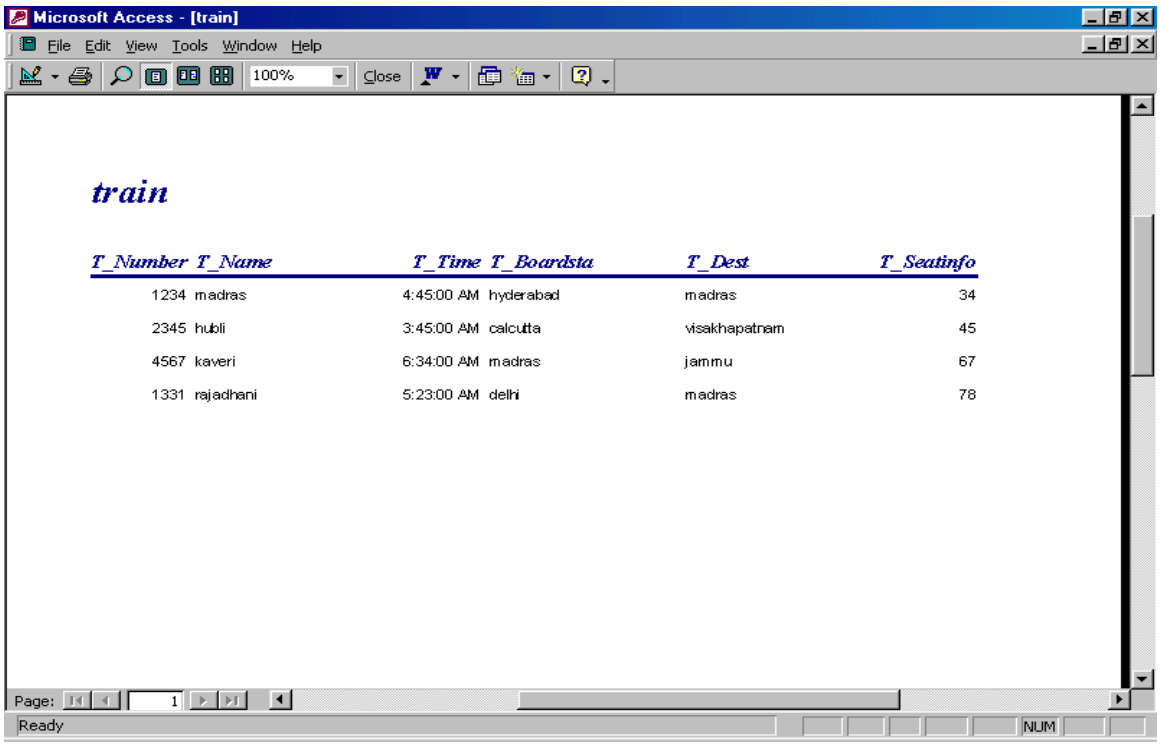
Train No	2076	Date of Journey	16-05-2006		
From	Hyderabad	To	New Delhi		
	Name of the Persons		Age	M/F	Requirement

This is the form that is to modified the train details. This form consists of no of seats textbox, which shows the remaining seats present in the train. The no of seats value is automatically changed when we create or cancel reservation. This is the form which consist of two buttons for the purpose of create reservation and to cancel reservation. When the seats text box has value zero than its, says no seats are remaining in the train to create reservation.

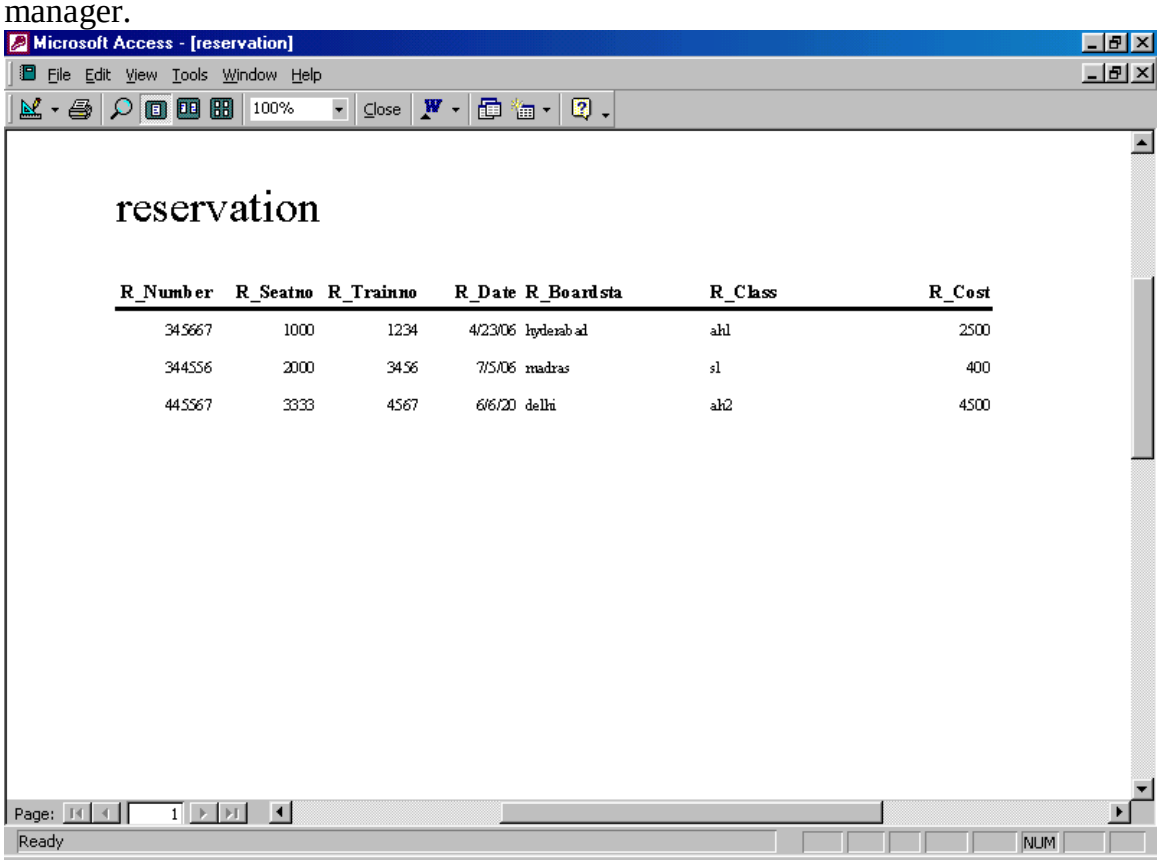


Reports for Railway Reservation System

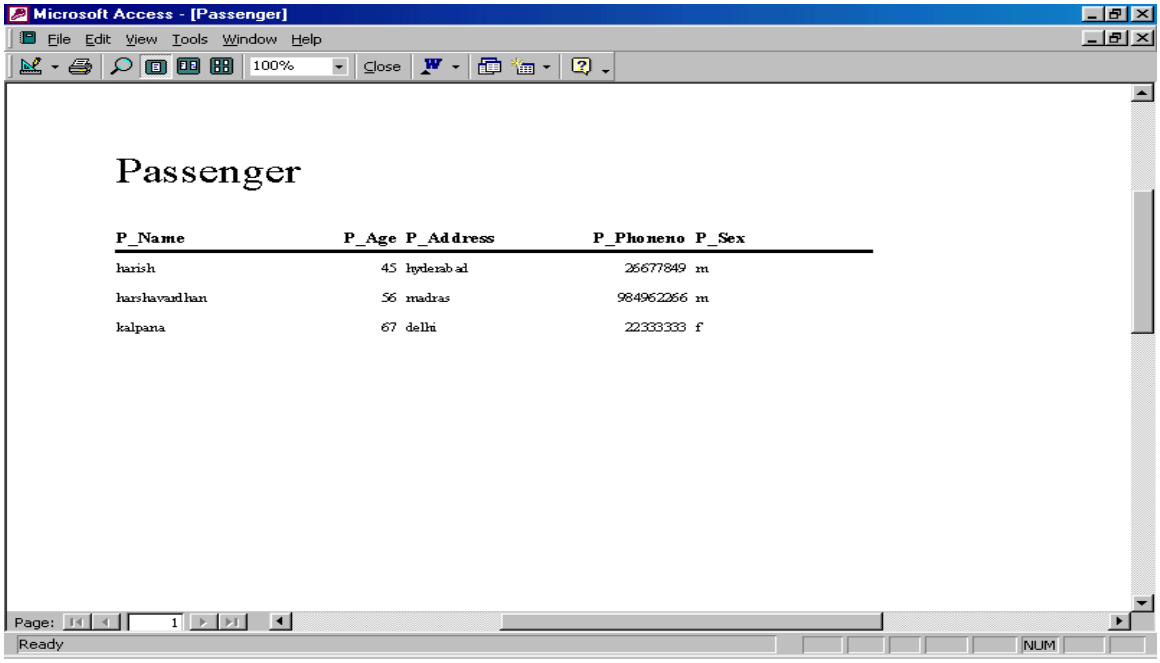
This is report generated by manager after modifying train details.



This is report generated by the clerk in order to give the reservation details to the manager.



This is report generated by the clerk about the passenger details who take the reservation.



Instructions for Security:

Security is an integral part of any system. Reservation clerk can create or cancel reservation when he goes through the login form. He is not able to update the train information and generating report. Manager update the train information and generate report only when he go through the login form. So, the system so secured, because of avoiding other people to update the train information and reservation details.

Data Model:

Class involved in project

- Passenger
- Manager
- Clerk
- Login
- Reservation
- Train
- Reports

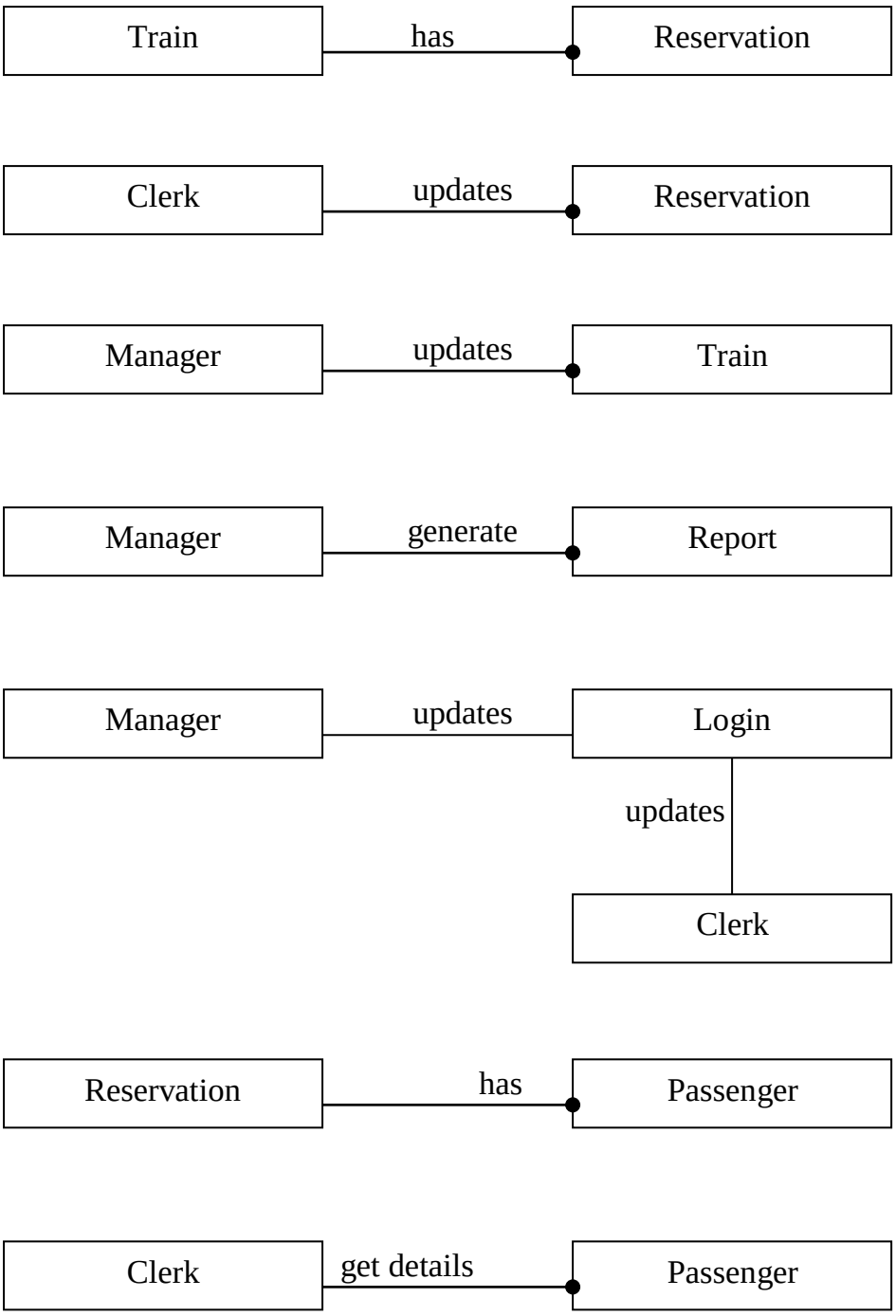
Attributes of the Entities are:

Objective Classes	Attribute
Passenger	P _ Name P _ Age P _ Address P _ Phoneno P _ Sex
Train	T _ Number T _ Name T _ Time T _ Boardsta T _ Dest T _ Seatinfo
Reservation	R _ Number R _ Seatno R _ Trainno R _ Date R _ Boardsta R _ Class R _ Cost
Clerk	C _ Username C _ Password
Manager	M _ Username M _ Password
Report	Re _ Number Re _ Name
Login	L _ Username L _ Password

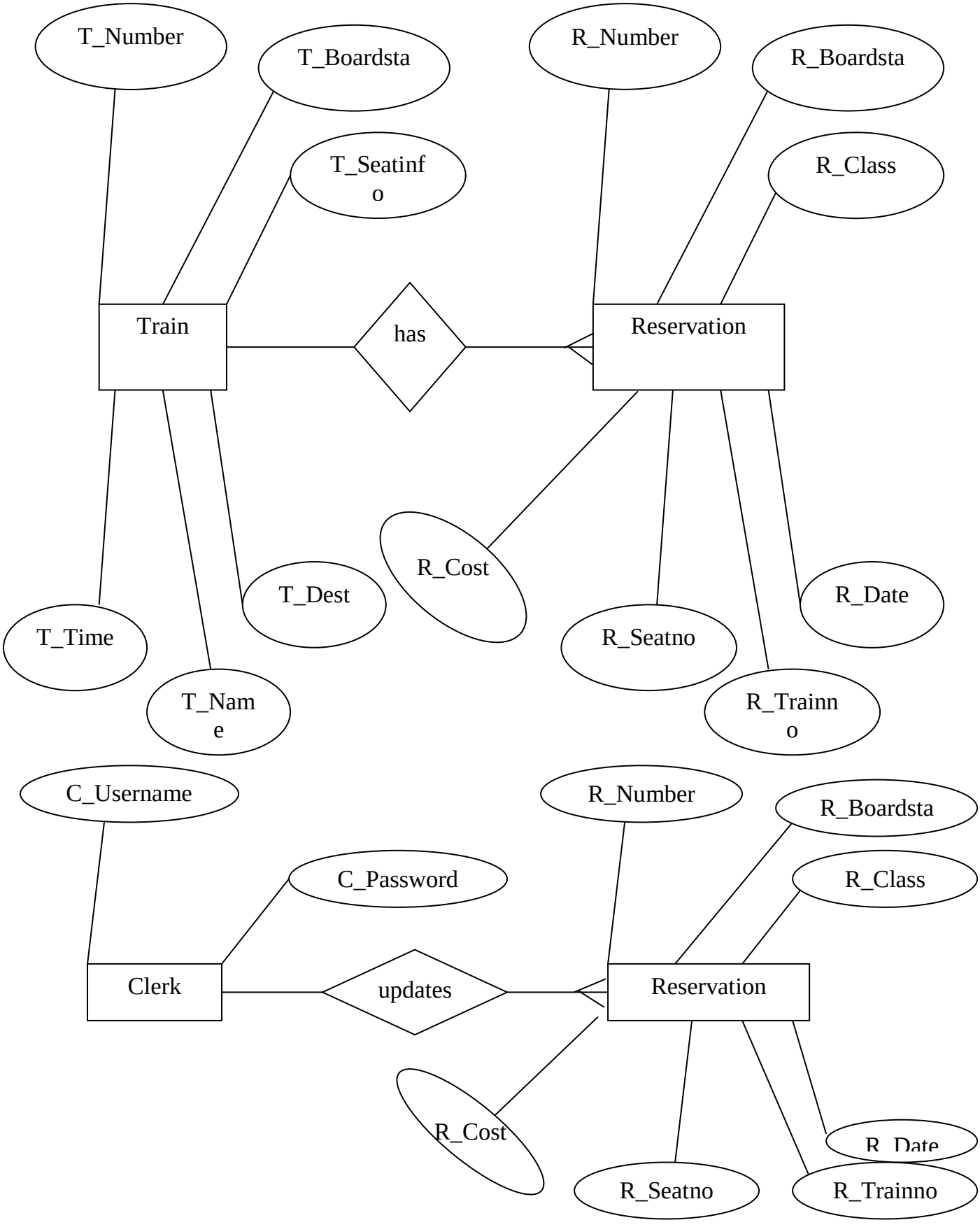
Association between the classes:

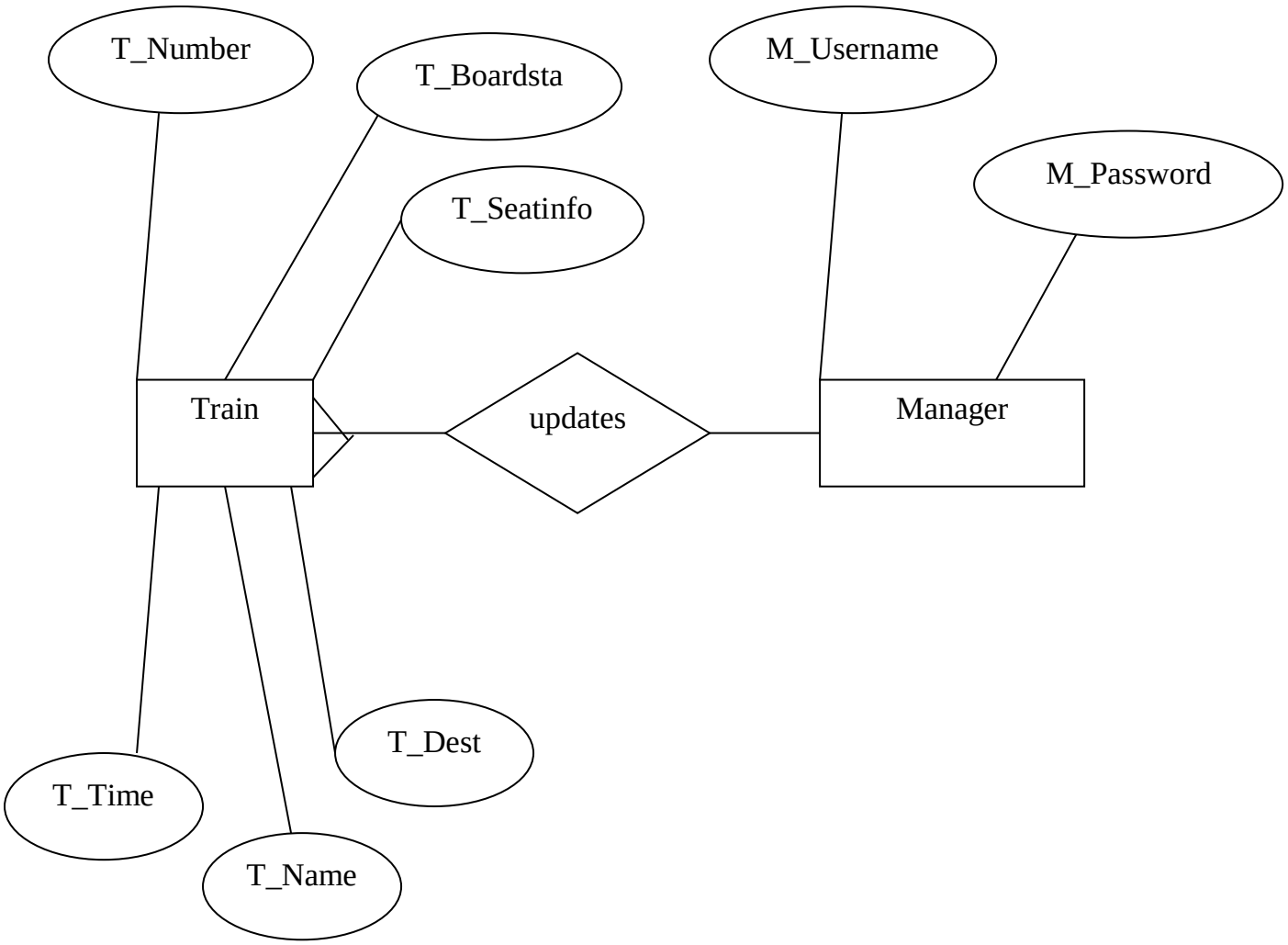
- Train has Reservation
- Clerk updates Login
- Clerk updates Reservation
- Manager updates Train
- Manager updates Login
- Manager generate Report
- Passenger has Reservation
- Clerk get details from Passenger

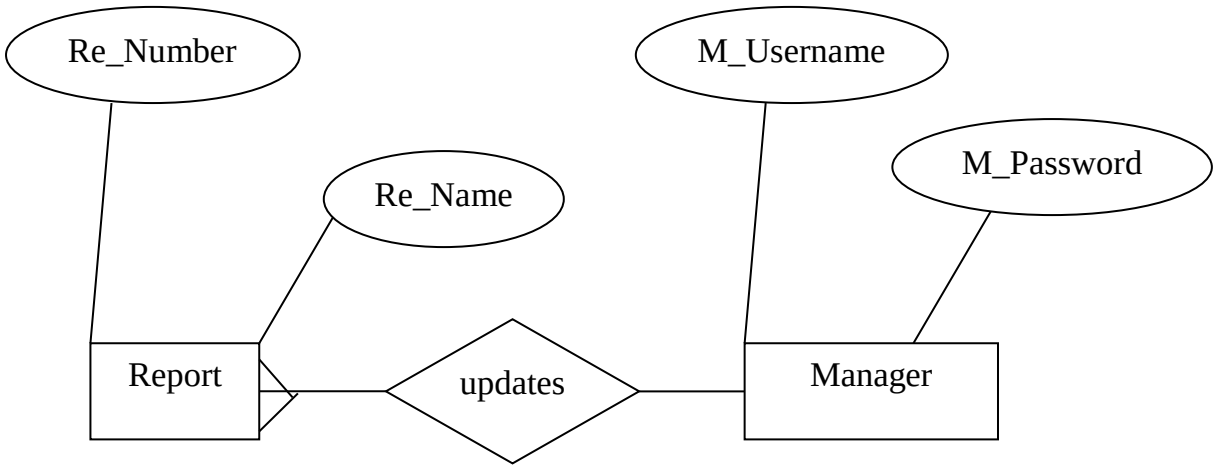
Object Diagrams for the System

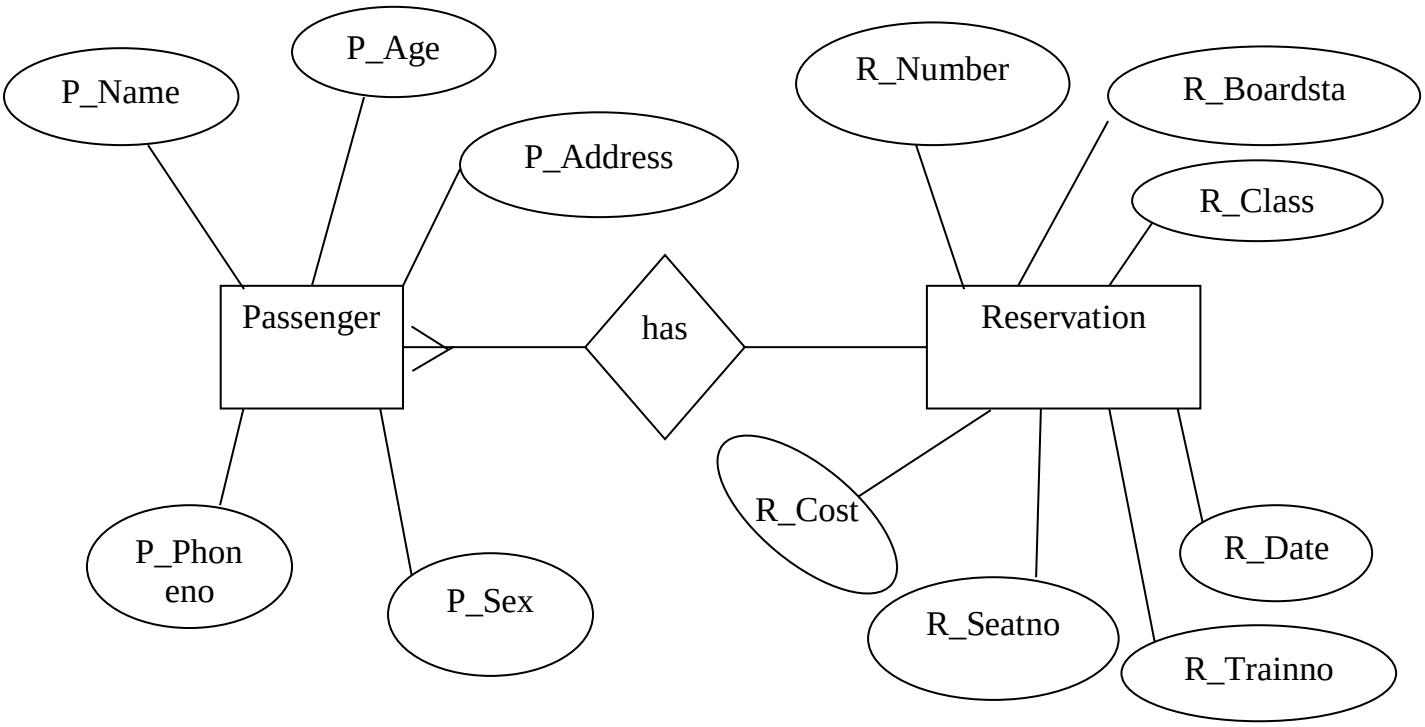


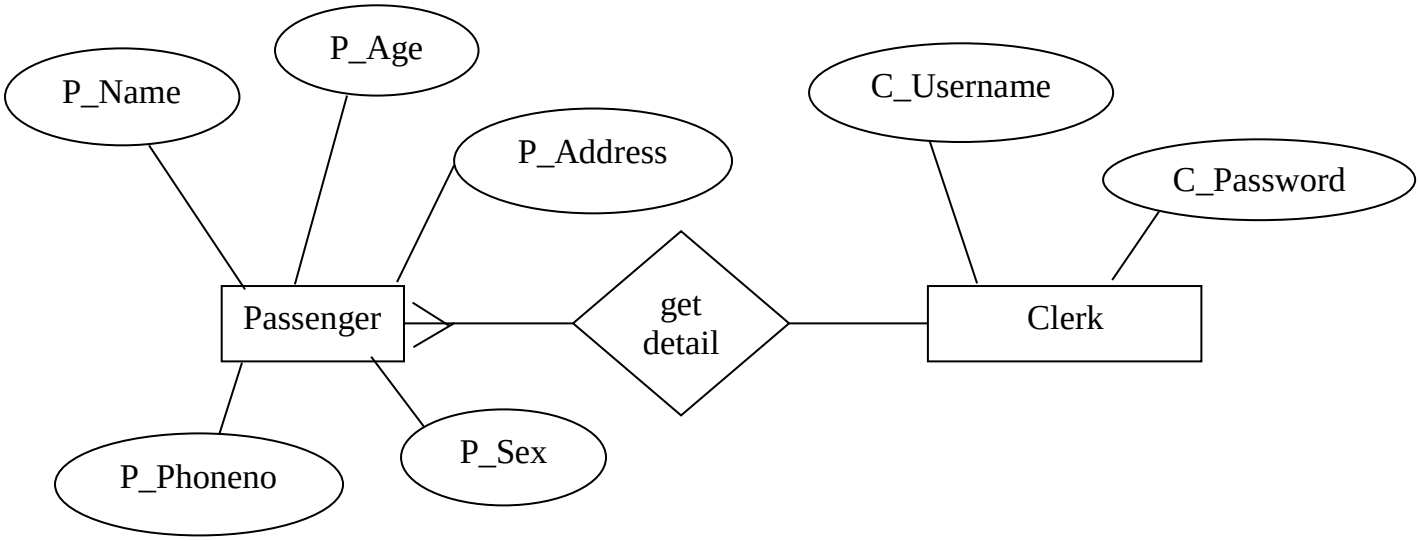
E-R Diagrams

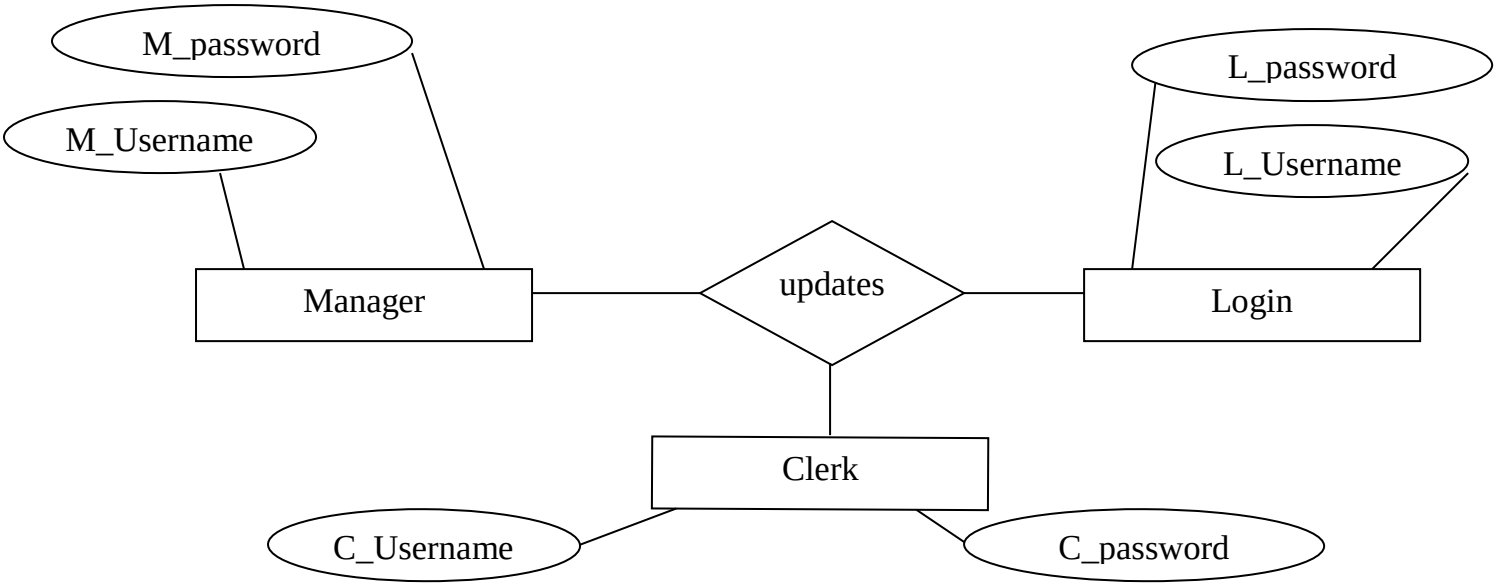

T_Number
T_Boardsta
T_Seatinf
o
R_Number
R_Boardsta
R_Class
Train
has
Reservation
T_Time
T_Nam
e
T_Dest
R_Cost
R_Seatno
R_Trainn
o
R_Date
C_Username
C_Password
Clerk
updates
R_Number
R_Boardsta
R_Class
Reservation
R_Cost
R_Seatno
R_Trainno
R_Date


T_Number
T_Boardsta
M_Username
M_Password
T_Seatinfo
Train
updates
Manager
T_Dest
T_Time
T_Name

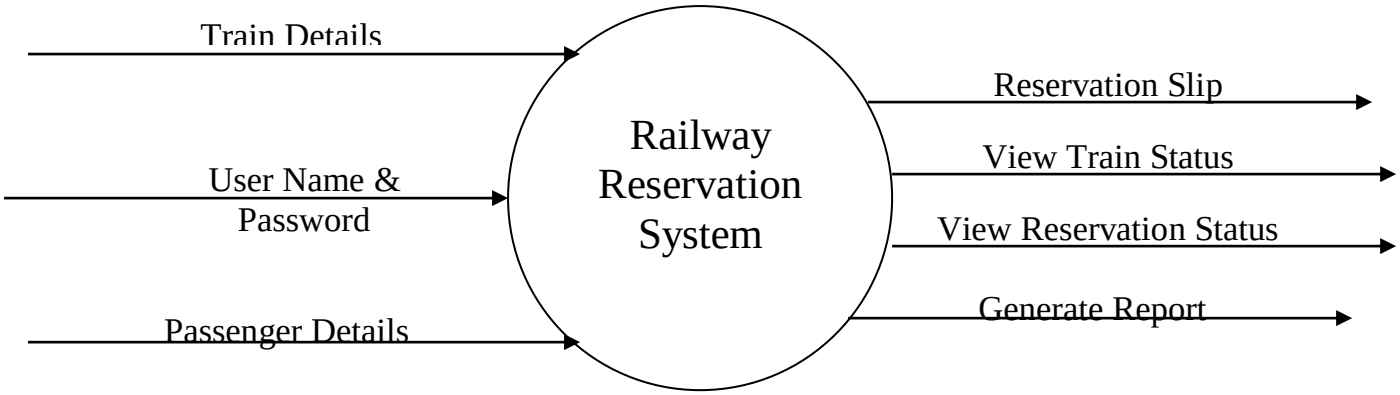

Re_Number
M_Username
M_Password
Re_Name
Report
updates
Manager

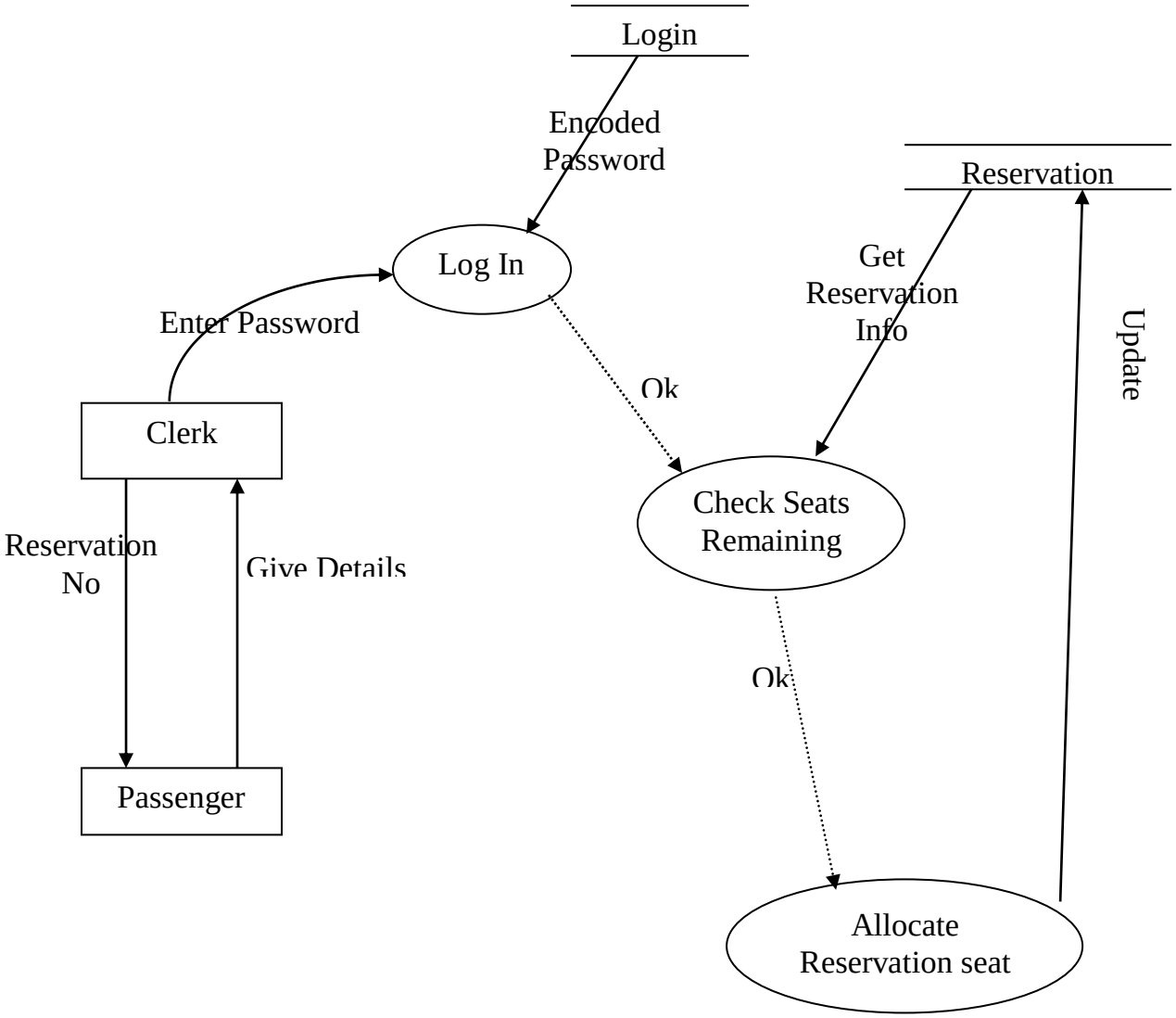

P_Name
P_Age
P_Address
Passenger
has
Reservation
R_Number
R_Boardsta
R_Class
P_Phon
eno
P_Sex
R_Cost
R_Seatno
R_Date
R_Trainno


P_Name
P_Age
P_Address
Passenger
get
detail
C_Username
C_Password
Clerk
P_Phoneno
P_Sex

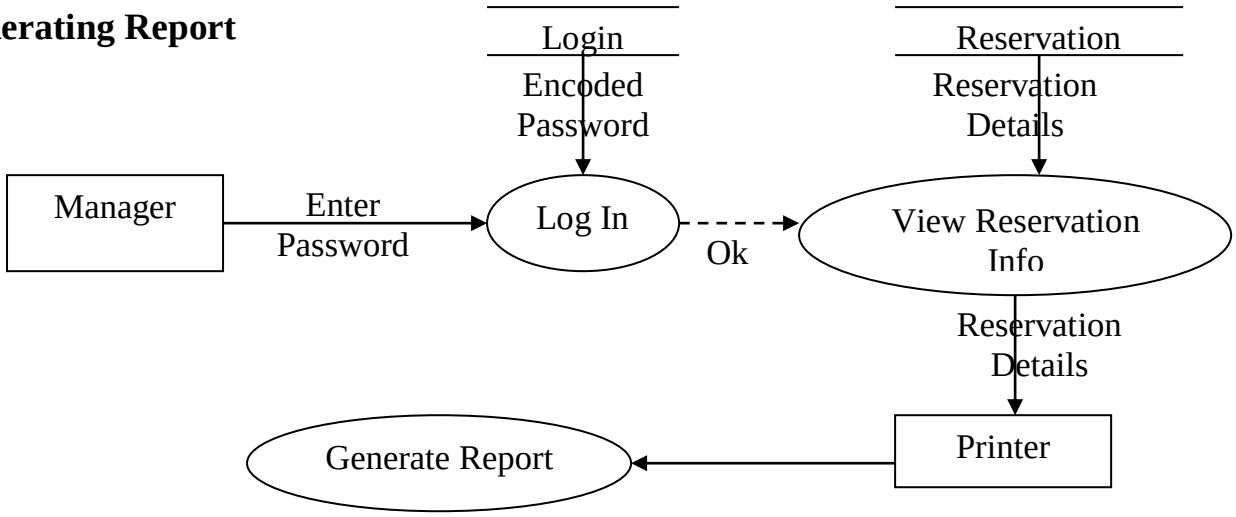

M_password
M_Username
L_password
L_Username
Manager
updates
Login
Clerk
C_Username
C_password

Functional Model

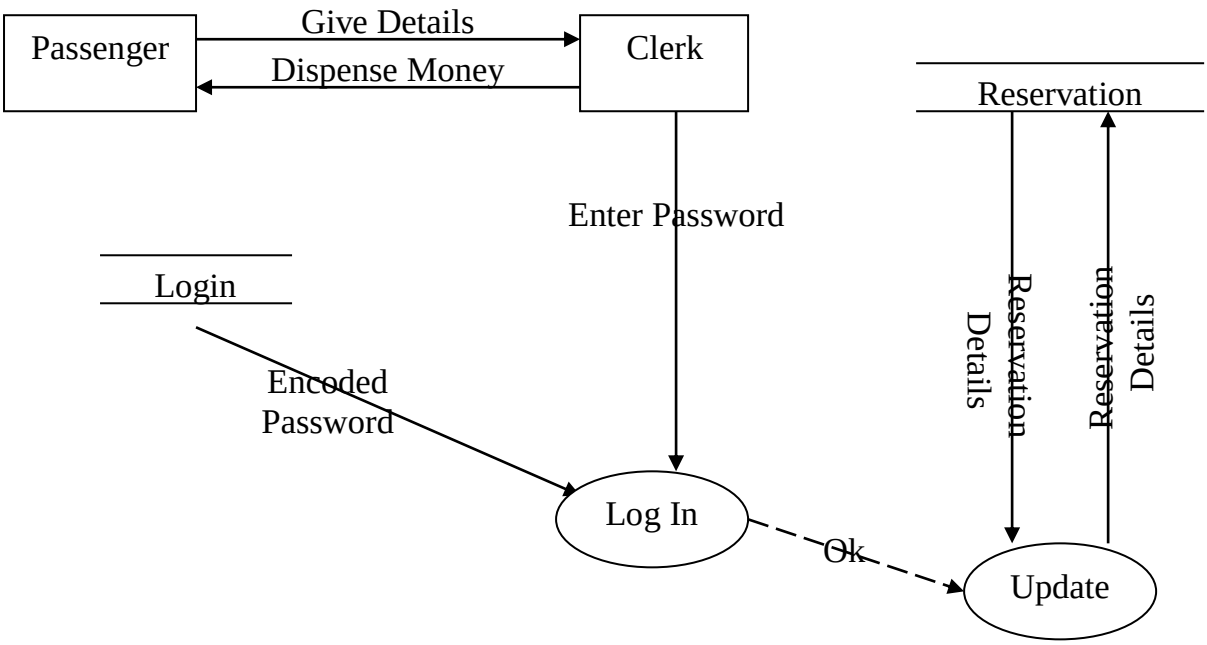




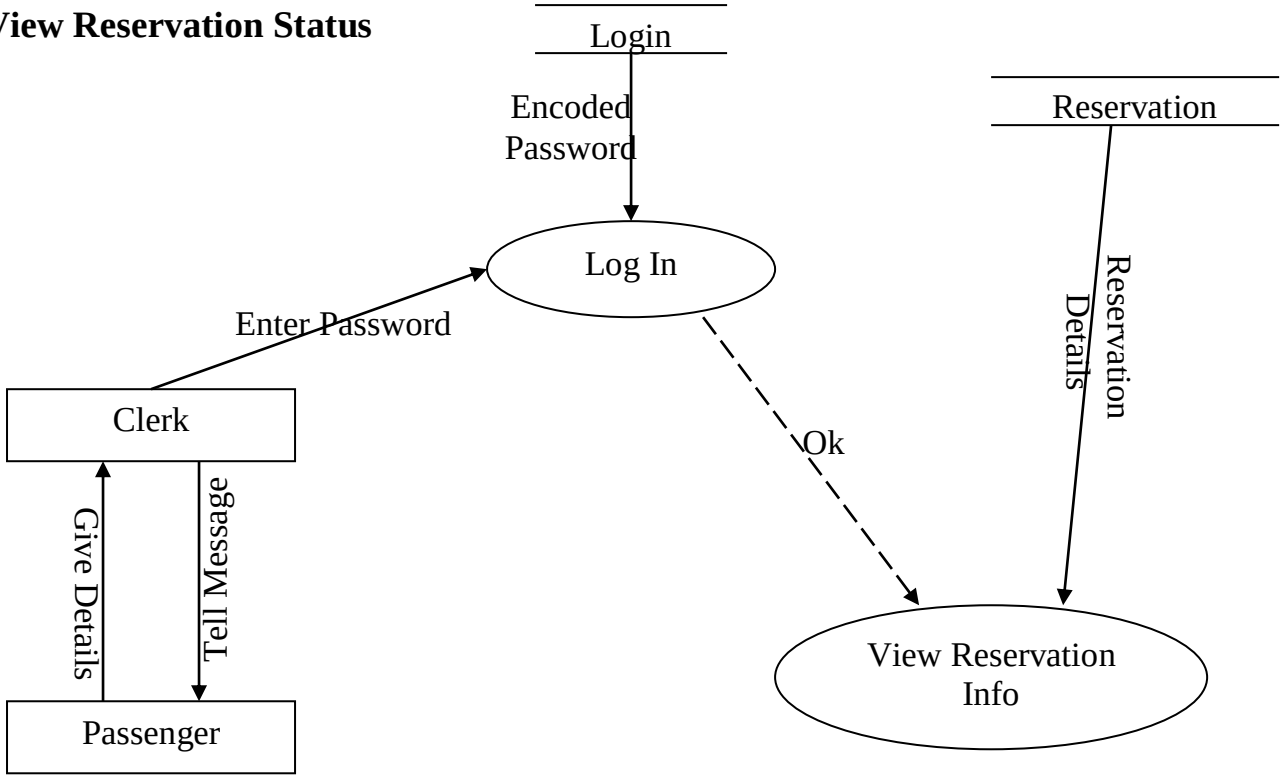
Generating Report



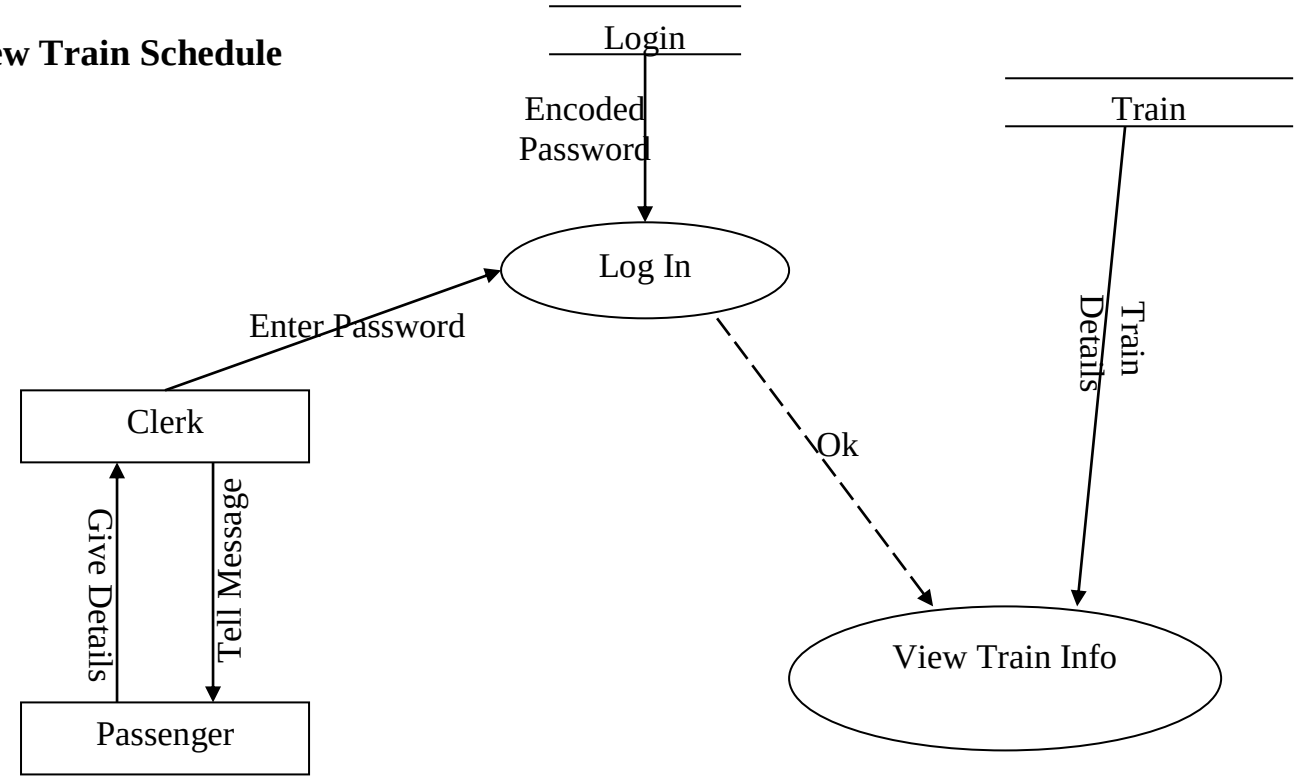
Cancel Reservation



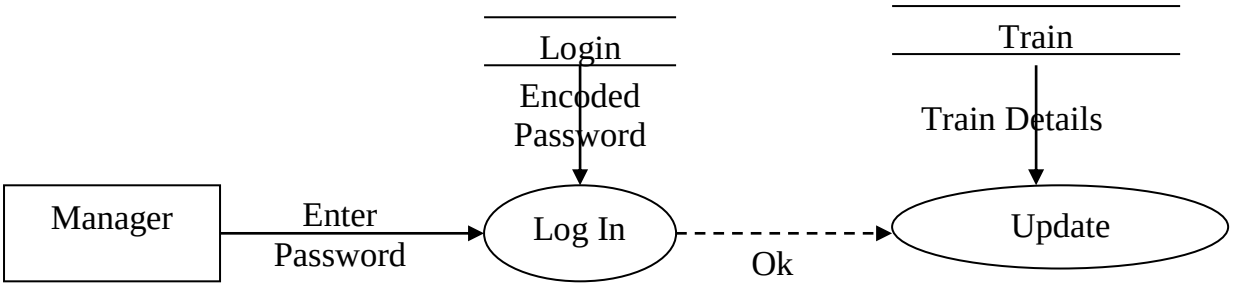
View Reservation Status



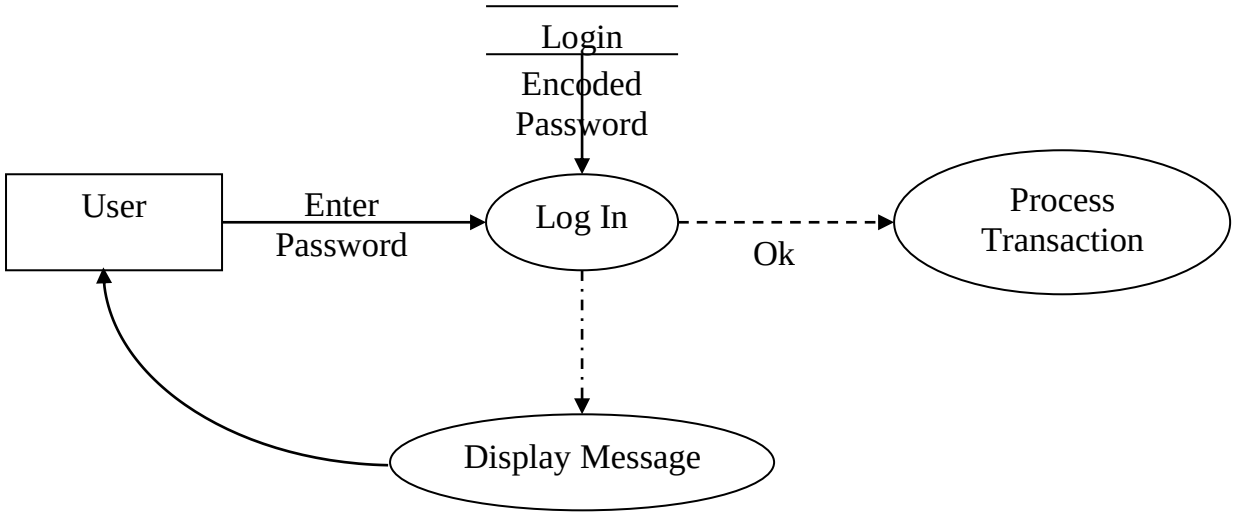
View Train Schedule



Update Train Info

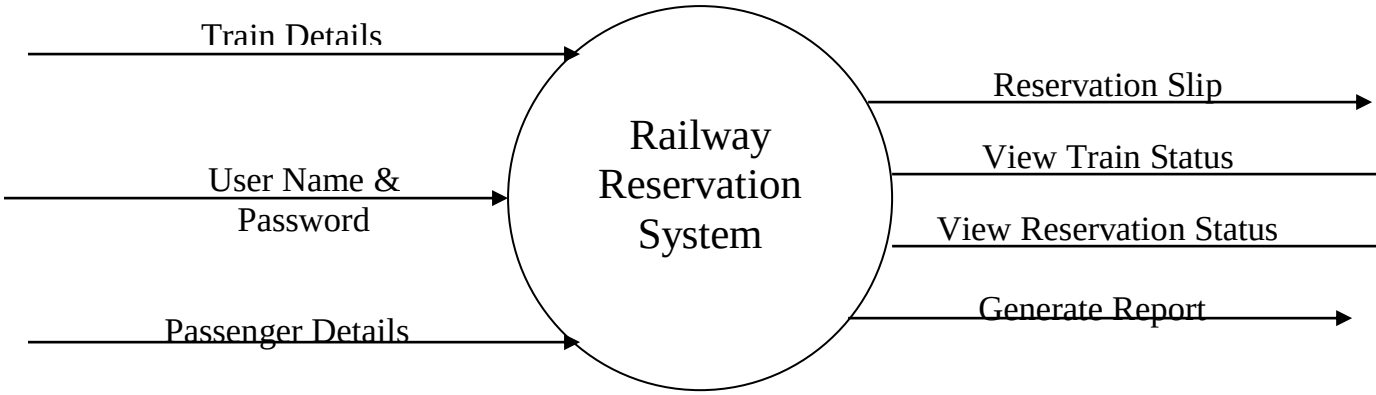


Verify login

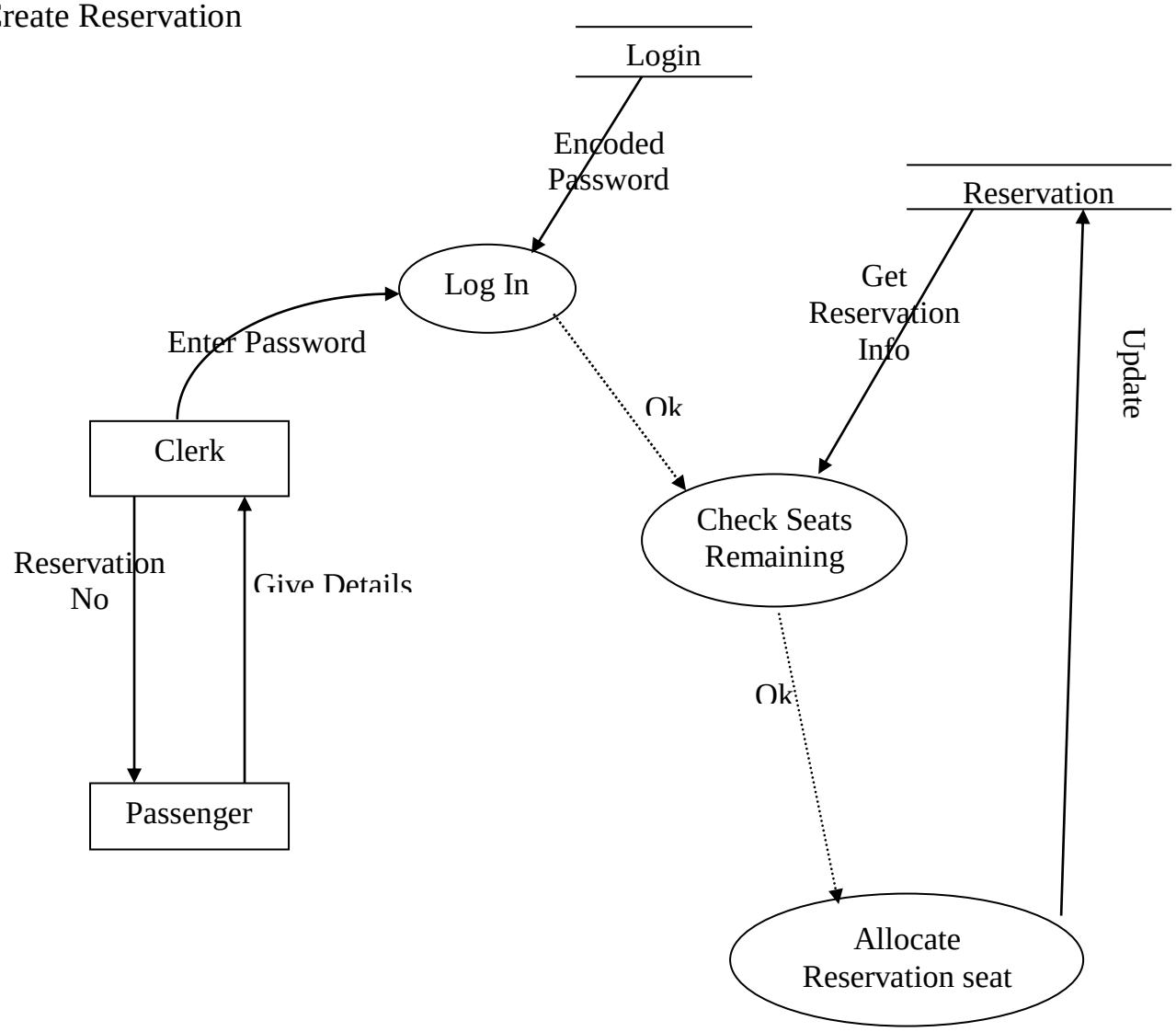


Session – 4

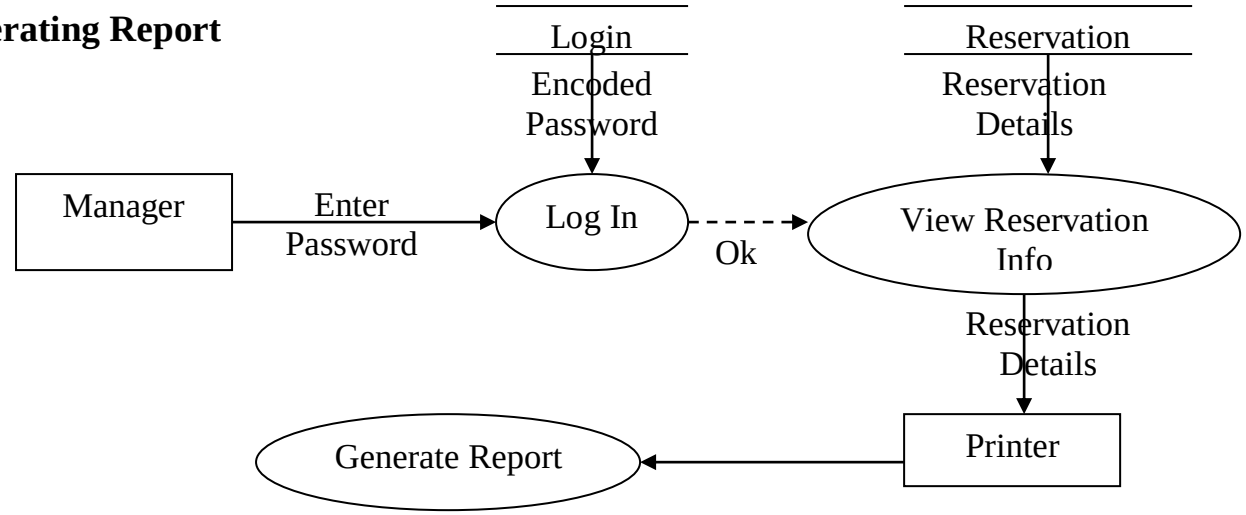
a) DFD's upto appropriate levels for Railway Reservation System



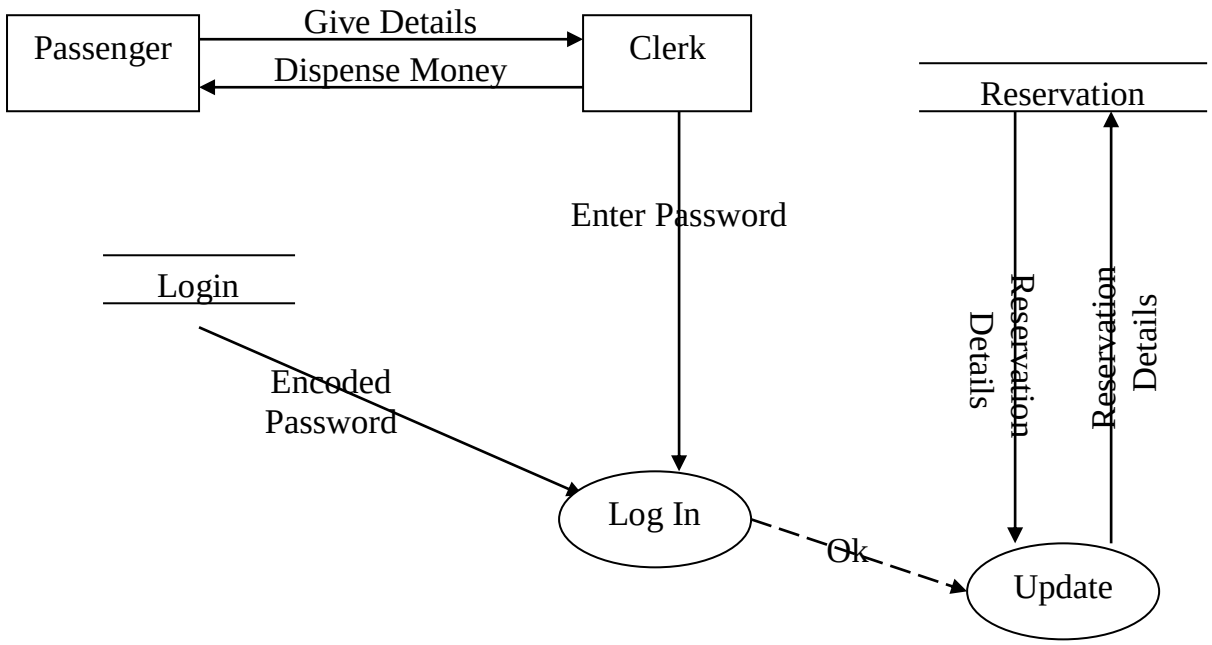
Create Reservation



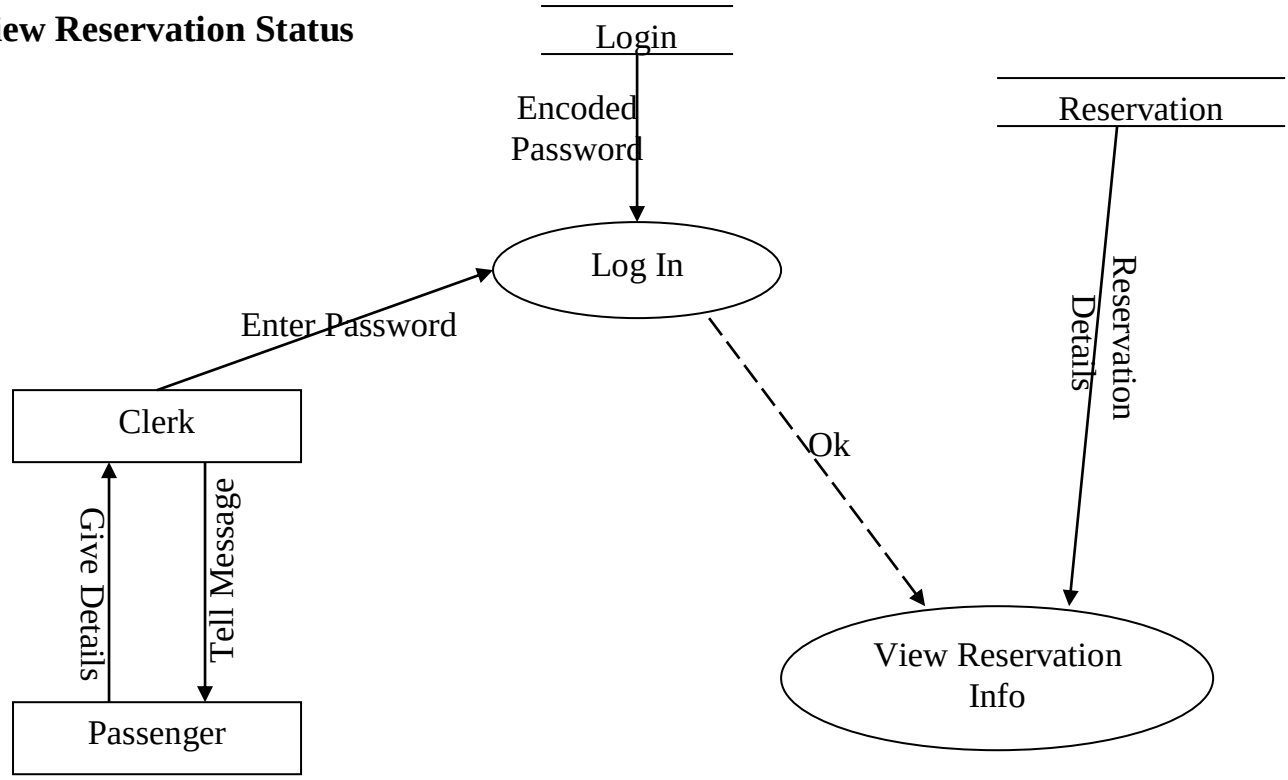
Generating Report



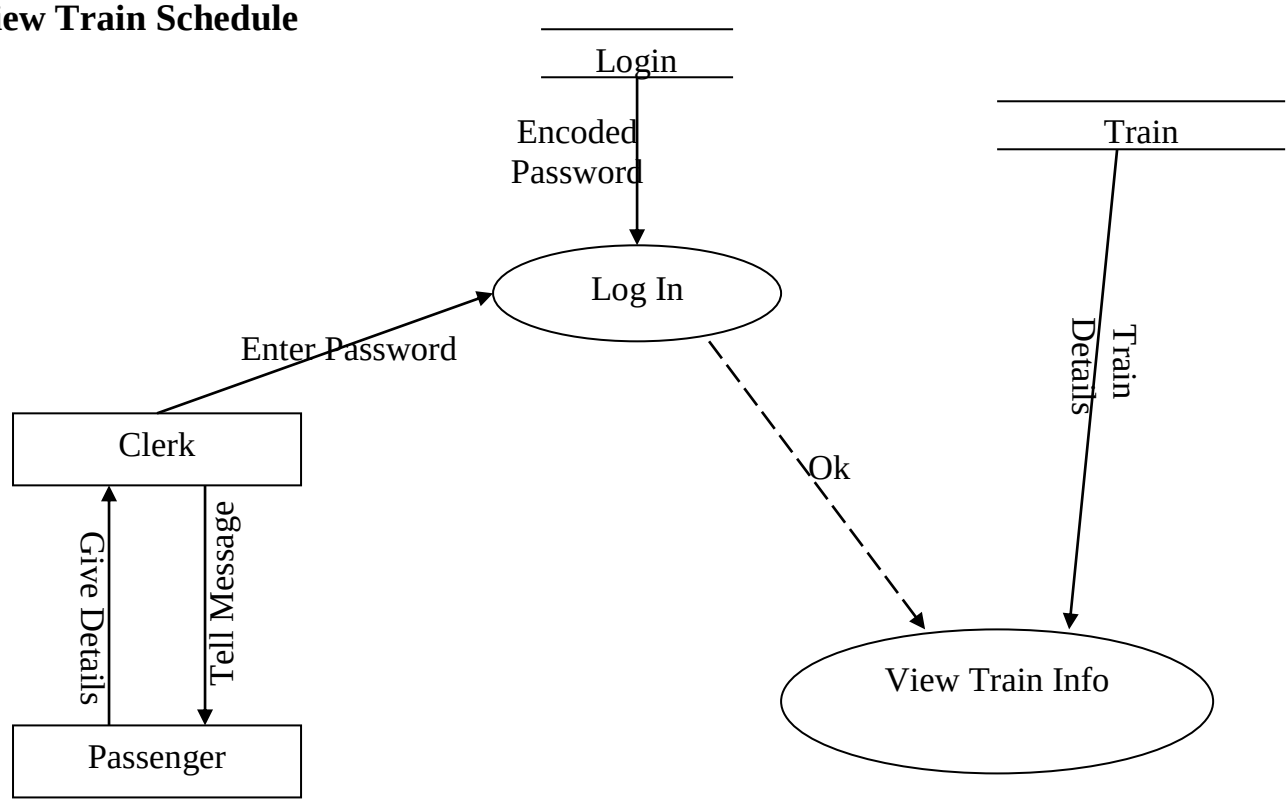
Cancel Reservation



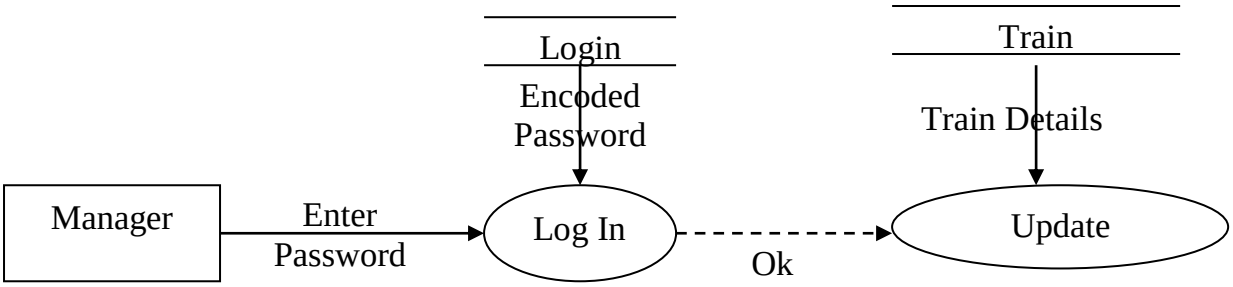
View Reservation Status



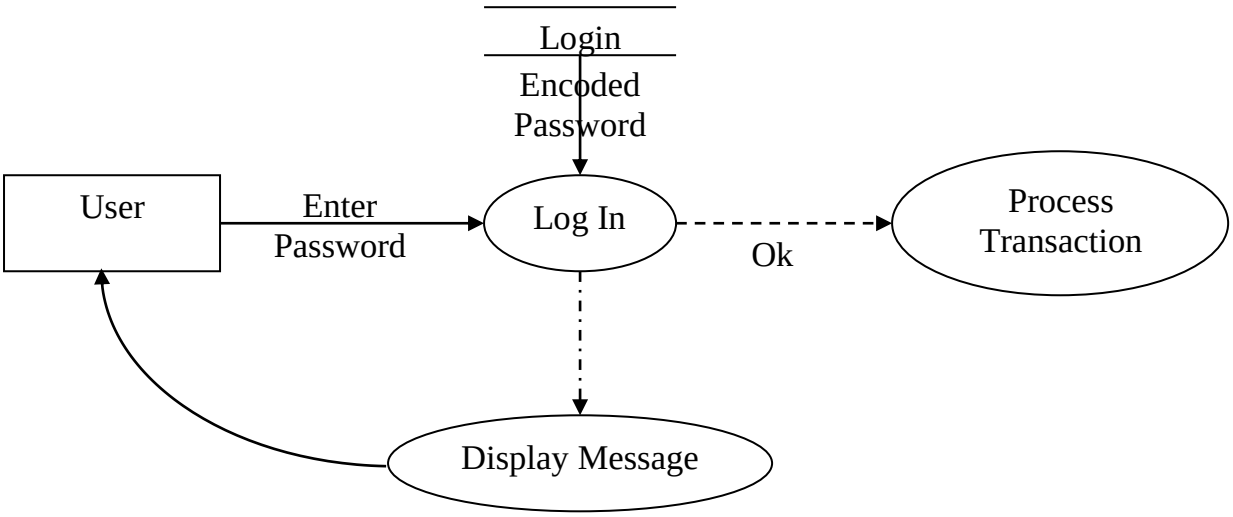
View Train Schedule



Update Train Info



Verify login



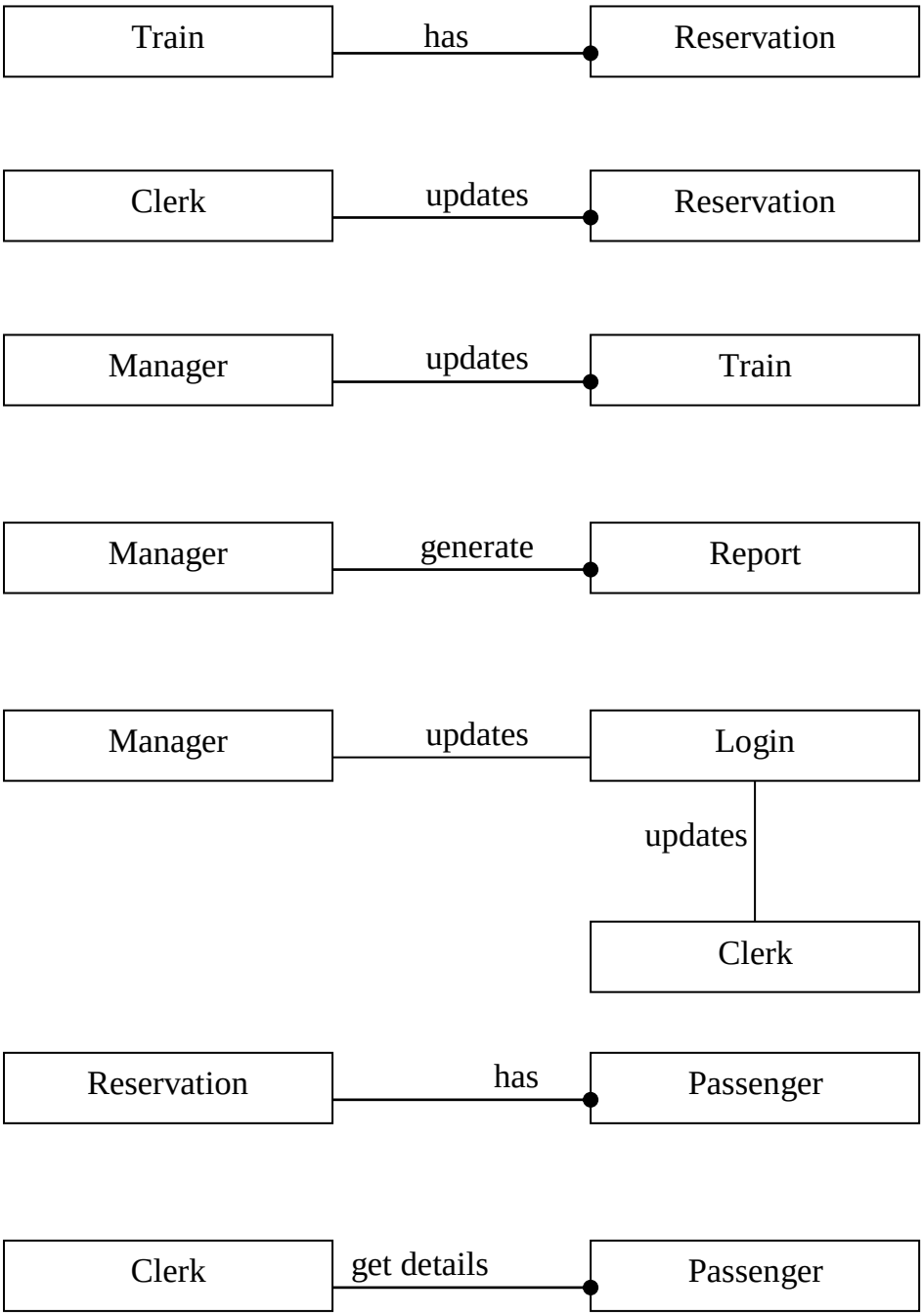
b) Draw ERD's for RRS. Describe the relationship between different entities.

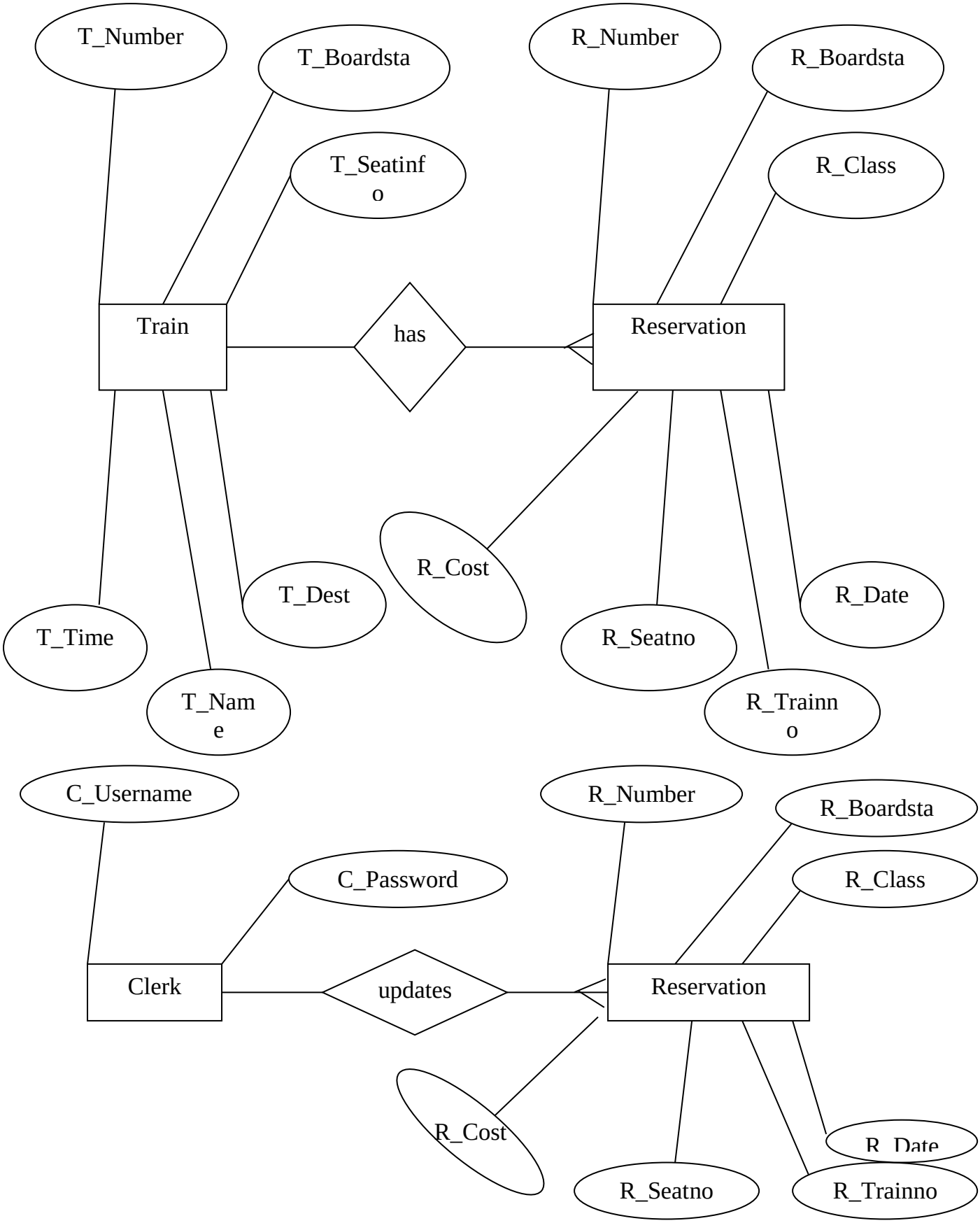
Attributes of the Entities are:

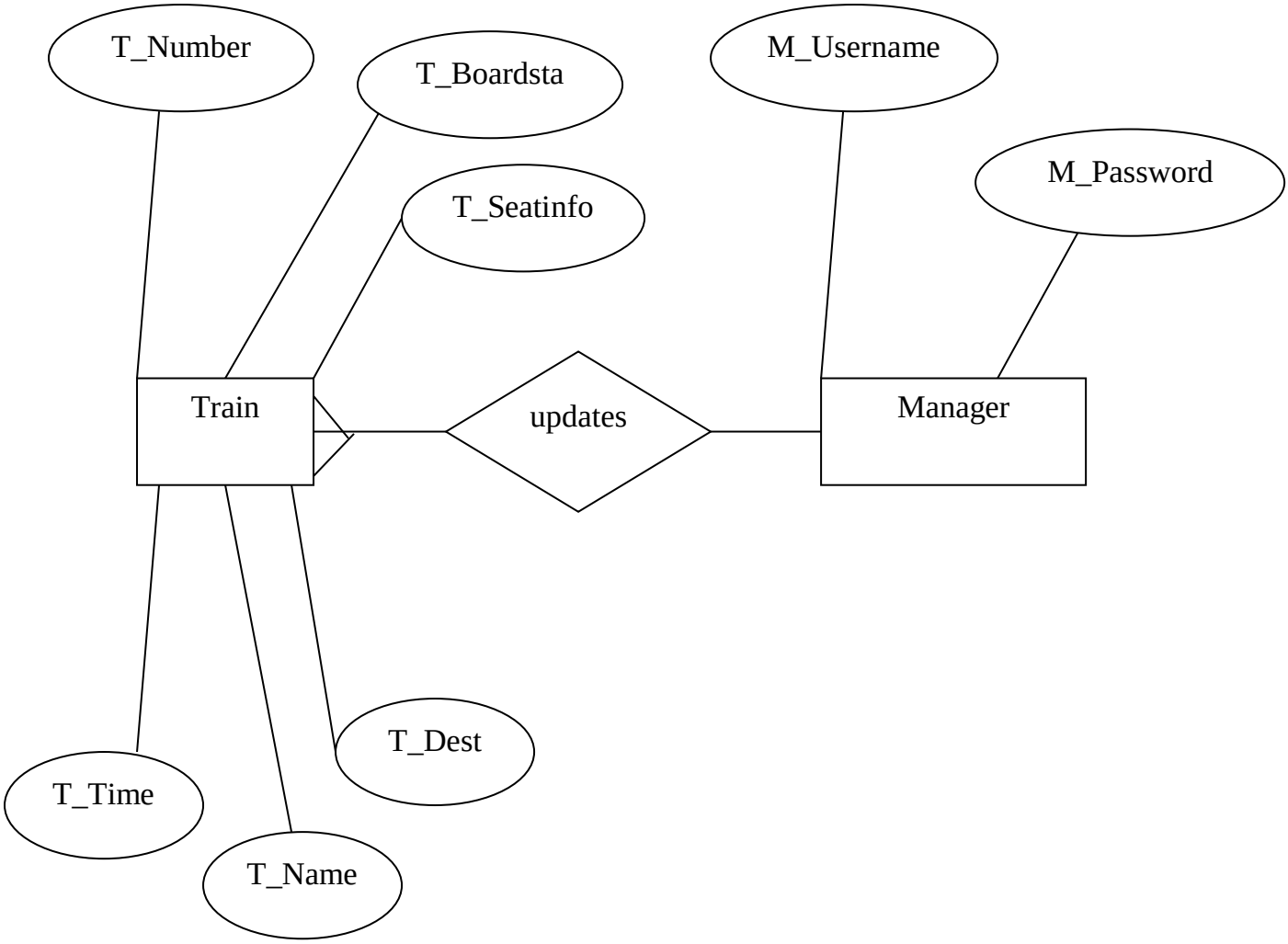
Objective Classes	Attribute
Passenger	P _ Name P _ Age P _ Address P _ Phoneno P _ Sex
Train	T _ Number T _ Name T _ Time T _ Boardsta T _ Dest T _ Seatinfo
Reservation	R _ Number R _ Seatno R _ Trainno R _ Date R _ Boardsta R _ Class R _ Cost
Clerk	C _ Username C _ Password
Manager	M _ Username M _ Password
Report	Re _ Number Re _ Name
Login	L _ Username L _ Password

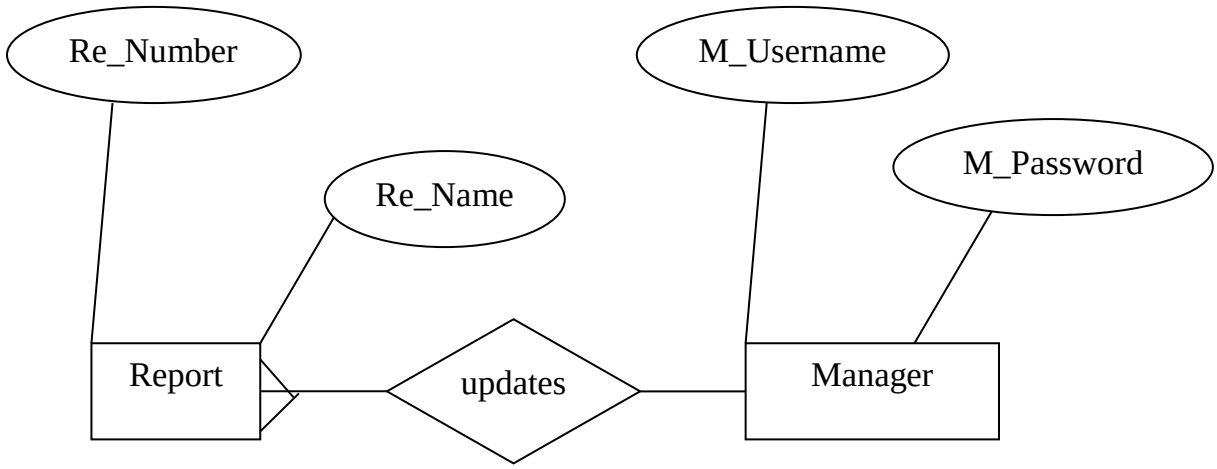
Association between the classes:

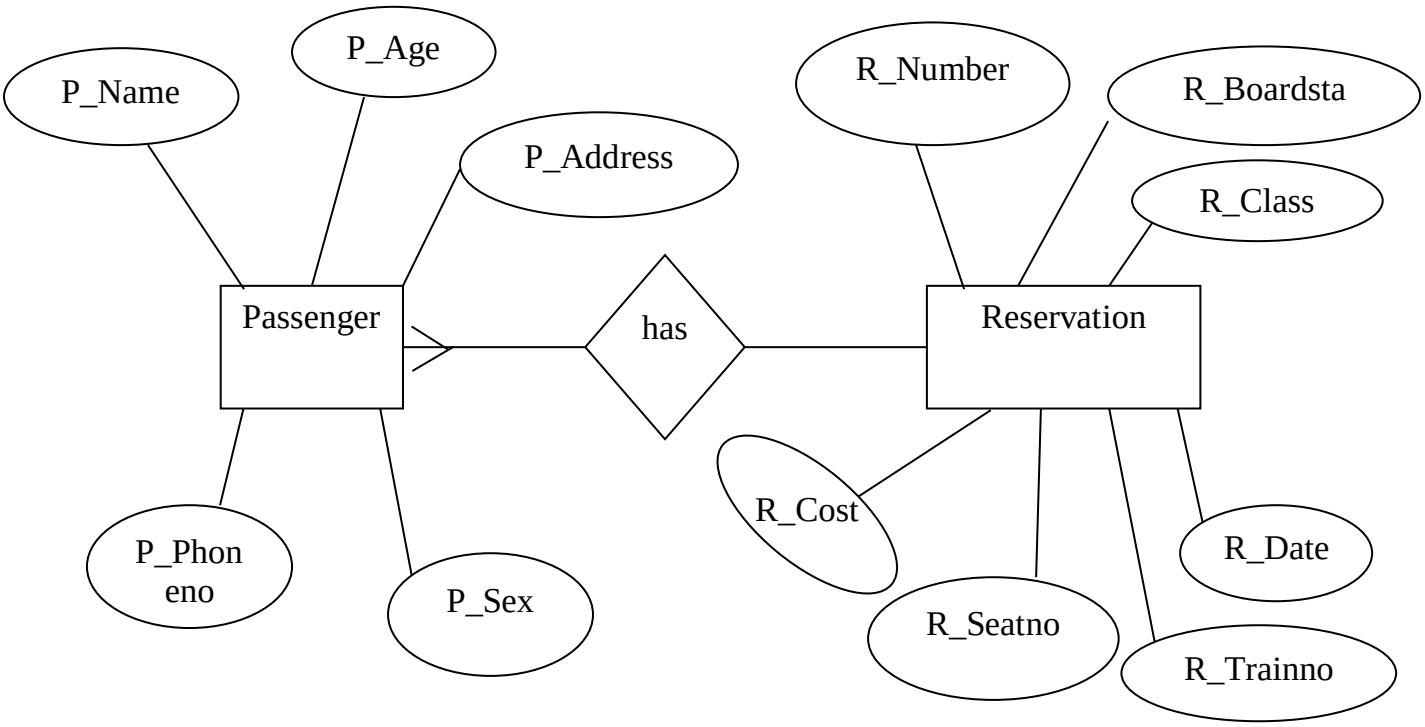
- Train has Reservation
- Clerk updates Login
- Clerk updates Reservation
- Manager updates Train
- Manager updates Login
- Manager generate Report
- Passenger has Reservation
- Clerk get details from Passenger

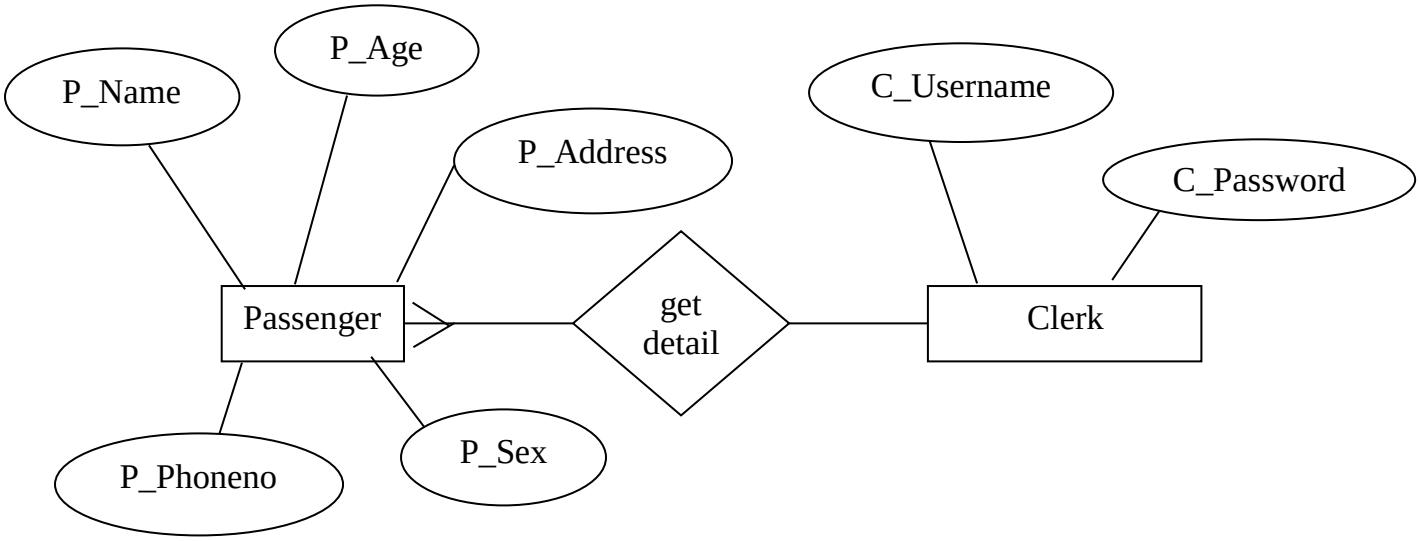

Train
has
Reservation
Clerk
updates
Reservation
Manager
updates
Train
Manager
generate
Report
Manager
updates
Login
updates
Clerk
Reservation
has
Passenger
Clerk
get details
Passenger

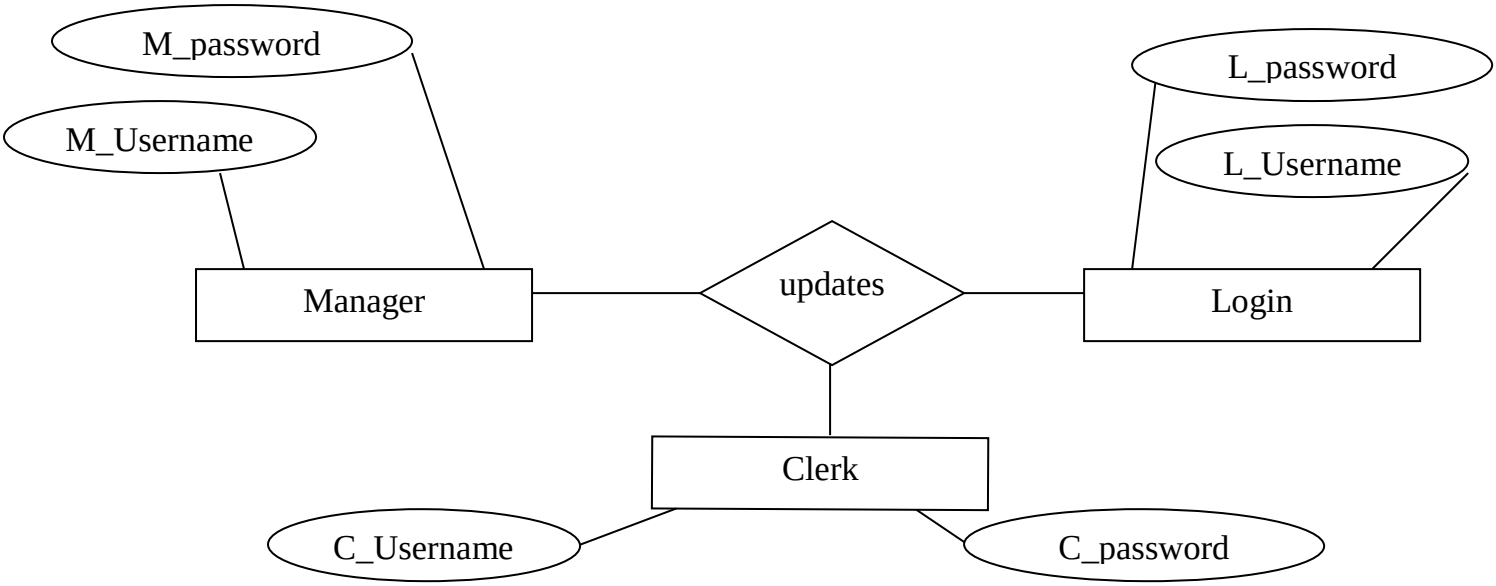

T_Number
T_Boardsta
T_Seatinf
o
R_Number
R_Boardsta
R_Class
Train
has
Reservation
T_Time
T_Nam
e
T_Dest
R_Cost
R_Seatno
R_Date
R_Trainn
o
C_Username
C_Password
Clerk
updates
R_Number
R_Boardsta
R_Class
Reservation
R_Cost
R_Seatno
R_Date
R_Trainno


T_Number
T_Boardsta
M_Username
M_Password
T_Seatinfo
Train
updates
Manager
T_Dest
T_Time
T_Name


Re_Number
M_Username
M_Password
Re_Name
Report
updates
Manager


P_Name
P_Age
P_Address
Passenger
has
R_Number
R_Boardsta
R_Class
Reservation
P_Phon
eno
P_Sex
R_Cost
R_Seatno
R_Date
R_Trainno


P_Name
P_Age
P_Address
Passenger
get
detail
C_Username
C_Password
Clerk
P_Phoneno
P_Sex



c) Design Data Dictionary for RRS.

Data Dictionary

The Dictionary of the terms that are used in railway reservation system is given in the below table:

Name	Expansion of Name	Where Used	Additional Description
Passenger		In Functional Diagrams & Data Diagrams	This is an object who wants to create, cancel reservation and to check reservation, train status.
Clerk	Reservation Clerk	In Functional Diagrams & Data Diagrams	This is an object who creates, cancel, and checks reservation status and update reservation info.
Admin	Manager	In Functional Diagrams & Data Diagrams	This is an object who maintains reservation and train info and generate report.
Printer		In Functional Diagrams & Data Diagrams	This is an object, which generate reports.
Login	Login information	In Functional Diagrams & Data Diagrams	This is a data base object, which stores the login password and user name.
Reservation	Reservation Information	In Functional Diagrams & Data Diagrams	This is a data base object, which stores the reservation details.

Train	Train Information	In Functional Diagrams & Data Diagrams	This is a data base object, which stores the train details.
Verify login		In Functional Diagrams	This is a process in order to check the user name and password given by the clerk and manager.
Update Train Info		In Functional Diagrams	This is a process in order to update the train information.
View Train Schedule		In Functional Diagrams	This is a process in order to view the train information.
View Reservation Status		In Functional Diagrams	This is a process in order to view the reservation details.
Create Reservation		In Functional Diagrams	This is a process in order to create reservation.
Cancel Reservation		In Functional Diagrams	This is a process in order to cancel reservation.
Generate Report		In Functional Diagrams	This is a process in order to generate report.
P _ Name	Passenger Name	In Data Diagrams	This is an attribute of a passenger.
P _ Age	Passenger Age	In Data Diagrams	This is an attribute of a passenger.
P _ Address	Passenger Address	In Data Diagrams	This is an attribute of a passenger.
P _ Phoneno	Passenger Phone No	In Data Diagrams	This is an attribute of a passenger.
P _ Sex	Passenger Sex	In Data Diagrams	This is an attribute of a passenger.
T _ Number	Train Number	In Data Diagrams	This is an attribute of a train. It is an unique field.
T _ Name	Train Name	In Data Diagrams	This is an attribute of a train.
T _ Time	Train Time	In Data Diagrams	This is an attribute of a train.
T _ Boardsta	Train Boarding Station	In Data Diagrams	This is an attribute of a train.
T _ Dest	Train Destination	In Data Diagrams	This is an attribute of a train.
T _ Seatinfo	Train Seat information	In Data Diagrams	This is an attribute of a train.
R _ Number	Reservation Number	In Data Diagrams	This is an attribute of a reservation. It is a unique attribute.
R _ Seatno	Reservation Seat Number	In Data Diagrams	This is an attribute of a reservation.

R _ Trainno	Reservation Train Number	In Data Diagrams	This is an attribute of a reservation, which is a foreign key for the 'Train Info' object.
R _ Date	Reservation Date	In Data Diagrams	This is an attribute of a reservation.
R _ Boardsta	Reservation Boarding Station	In Data Diagrams	This is an attribute of a reservation.
R _ Dest	Reservation Destination	In Data Diagrams	This is an attribute of a reservation.
R _ Class	Reservation Class	In Data Diagrams	This is an attribute of a reservation.
R _ Cost	Reservation Cost	In Data Diagrams	This is an attribute of a reservation.
C _ Username	Clerk User Name	In Data Diagrams	This is an attribute of a clerk
C _ Password	Clerk Password	In Data Diagrams	This is an attribute of a clerk
M _ Username	Manager User Name	In Data Diagrams	This is an attribute of a manager.
M _ Password	Manager Password	In Data Diagrams	This is an attribute of a Manager
Re _ Number	Report Number	In Data Diagrams	This is an attribute of a report.
Re _ Name	Report Name	In Data Diagrams	This is an attribute of a report.
L _ Username	Login User Name	In Data Diagrams	This is an attribute of a login.
L _ Password	Login Password	In Data Diagrams	This is an attribute of a login.

Session – 5

The Railway Reservation System consists of so many interfaces in order to input and output the data. As this Railway Reservation System must involves the transfer of data to the server. The Railway Reservation System must manage the server in order to store the data of the reservation, train details. So, we must the data of the reservation, train to the server. So, we choose the Java Programming for the Railway Reservation System, because it is very suitable to the Client-Server architecture. Java programming is very efficient in interface designing. The below module is to create the interface to enter the train details and to create, cancel the reservation. The data that is entered in the text boxes, is stored in the database. When we click the " Create Reservation" button, then we get another interface to enter the reservation details.

When we click the " Create Reservation" button, then we get another interface to enter the reservation details for cancel .The data that is entered in the Create Reservation form can automatically updates the train details. The data that is entered in the Cancel Reservation form can also automatically updates the train details. The data is automatically stored in the server. Remaining users who are using that server can change this data.

```
class myframe extends JFrame implements ActionListener
{
JLabel l1,l2,l3,l4,l5,l6,l7;
JTextField t1,t2,t3,t4,t5;
JList lt1;
Jpanel p1,p2,p3,p4,p5,p6,p7;
JButton  b1,b2,b3,b4;
Container c;
Myframe(String name)
{
super(name);
c=getContentPane();
c.setLayout(new FlowLayout(FlowLayout.LEFT,30,30));
p1=new Jpanel();
p2=new Jpanel();
p3=new Jpanel();
p4=new Jpanel();
p5=new Jpanel();
p6=new Jpanel();
l1=new Jlabel("Train No. :",JLabel.RIGHT);
l2=new Jlabel("Train Name. :",JLabel.RIGHT);
l3=new Jlabel("Train Time :",JLabel.RIGHT);
l4=new Jlabel("Train BoardStation :",JLabel.RIGHT);
l5=new Jlabel("Train Destination :",JLabel.RIGHT);
lt1=new JList(6);
l6=new Jlabel("Train Seatinfo :",JLabel.RIGHT);
```

```
t1=new JTextField(30);
t2=new JTextField(30);
t3=new JTextField(30);
t4=new JTextField(30);
t5=new JTextField(30);
b1=new JButton(“Create Reservation”);
b2=new JButton(“Cancel Reservation”);
b3=new JButton(“Save”);
b4=new JButton(“Cancel”);
p1.add(l1);
p1.add(lt1);
p2.add(l2);
p2.add(t1);
p3.add(l3);
p3.add(t2);
p4.add(l4);
p4.add(t3);
p5.add(l5);
p5.add(t4);
p6.add(l6);
p6.add(t5);
p7.add(b1);
p7.add(b2);
c.add(p1);
c.add(p2);
c.add(p3);
c.add(p4);
c.add(p5);
c.add(p6);
c.add(p7);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
String msg=ae.getActionCommand();
if(msg.equals(“Save”))
{
Class.for(“c://mydocuments.”);
Connection conn=DriverManager.getConnection(“harish”,1234567);
conn.getAutoCommit();
PreparedStatement pstmt=conn.prepareStatement(“insert into train
values(&T_Number,’&T_Name’,&T_Time,&T_Boardsta,&T_Dest,&T_Seatinfo)”
);
```

```
pstmt.setValue(1, lt1.getSelectedItem());
pstmt.setValue(2, t1.getText());
pstmt.setValue(3, t2.getText());
pstmt.setValue(4, t3.getText());
pstmt.setValue(5, t4.getText());
pstmt.setValue(6, t5.getText());
}
if(msg.equals("Cancel"))
{
JoptionPane.showMessageDialog(null,"The values cannot be entered");
f.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
}
if(msg.equals("Create Reservation"))
{
myframe2.setVisible(true);
f.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
//_ _ _ _
}
if(msg.equals("Create Reservation"))
{
if(Integer.parseInt(t5.getText())==0)
JOptionPane.showMessageDialog(null,"Cannot have seats");
else
{
myframe3.setVisible(true);
f.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
//_ _ _ _
}
}
}
public static void main(String args[])
{
myframe f=new myframe("Train Details");
f.setVisible(true);
f.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
}
}
```

Inorder to view the reservation details, the module code is:

```
Statement stmt=conn.createStatement();
ResultSet rst=stmt.updateQuery("select * from reservation");
While(rst.next())
{
t1.setText(Integer.toString(rst.getText()));
//**************************
```

}
In the above program we have to maintain constraints, when the TextBox t5 has value 0 then we do not create reservation. If we create reservation at this situation, then it automatically generates error message. The other peoples except manager cannot update train details, the train details is highly secured.
So, we have to maintain login form to update train details.
For efficient design of interfaces, we use Swings in Java. In modular design we have to concentrate on two concepts inorder to design efficient modules. The two concepts are,

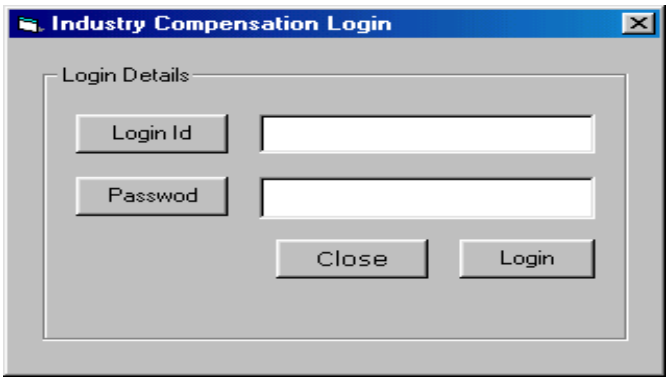
- Coupling
- Cohesion

We have to design modules with less coupling and with high cohesion. Coupling specifies the relation between two modules. It specifies how the modules interact with each other. There must be less relation between the modules. Because, when we change one module, then we have to change another module when there is high relation between them. If we have less coupling between the modules then it is easy to modify the software product.

Cohesion represents the functionality of the function. It specifies how the instructions in the module can perform the functionality of the function. We have to maintain high cohesion between modules. Because, all instructions in the module must perform single functionality. So, mainly we have to concentrate on the Coupling and Cohesion.

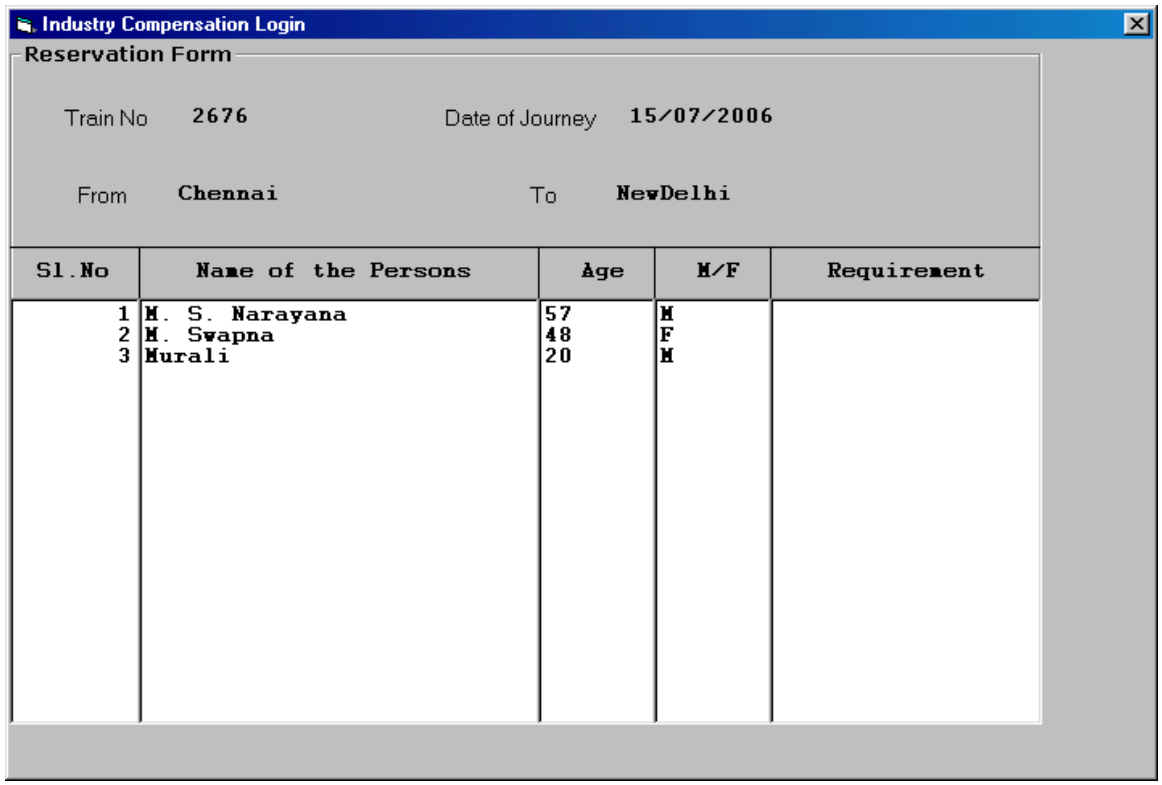
Session – 6

The below form is used to enter the user to logon to the system. In this GUI, there are two text boxes in which we have to enter the login id and password. The system checks the values that are entered in text boxes with the Login Id and Password, which is stored in database. If the given values are correct then only the user will enter into RRS system. Otherwise, a message will be appeared on the screen.



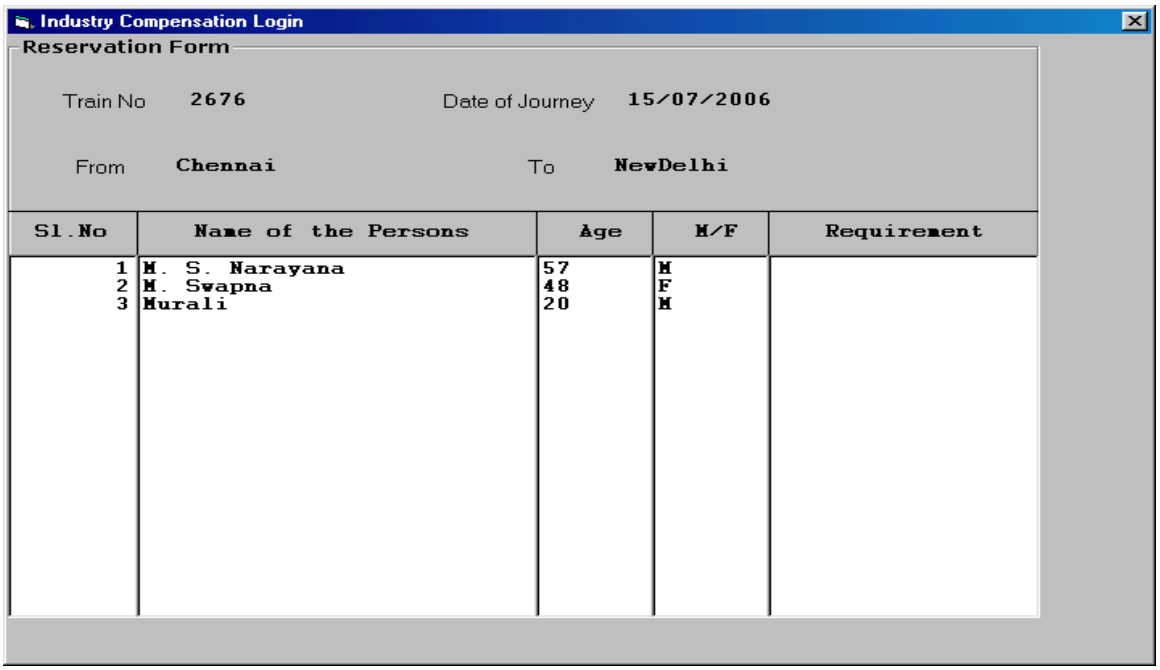
This is a message box that is generated when we entered wrong password.

This is a form, which is to enter Reservation Details. This form consists of four text boxes, which is to enter details from user in order to create reservation for the user. The clerk cannot create reservation more than the seats present in the train. If he wants to create reservation more than that then a message will be appeared on the screen.

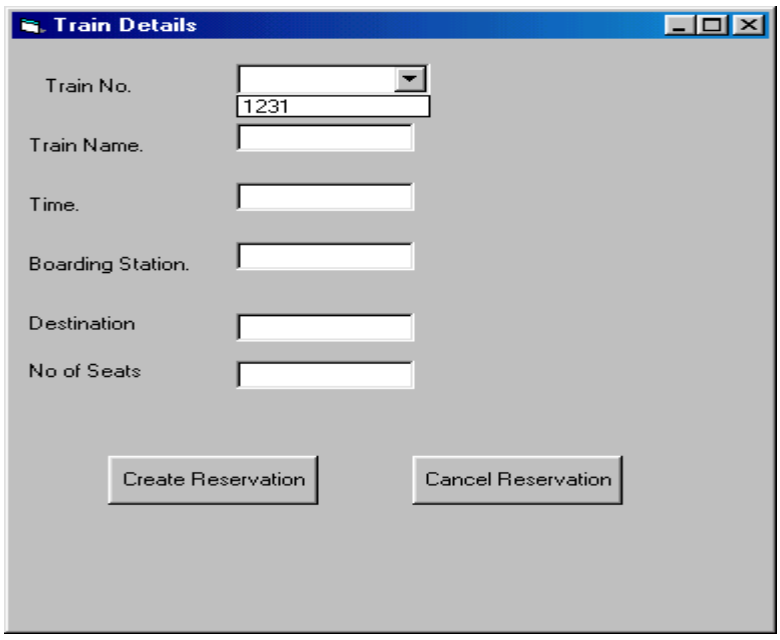


This is a message box that is generated when clerk creates the reservation more than the seats in the train.

This is the reservation slip that is given to the passenger after creating the reservation. This consists of the details of passenger and their seat numbers in which they are traveling. This is consists of train details, date of journey and boarding station, destination.

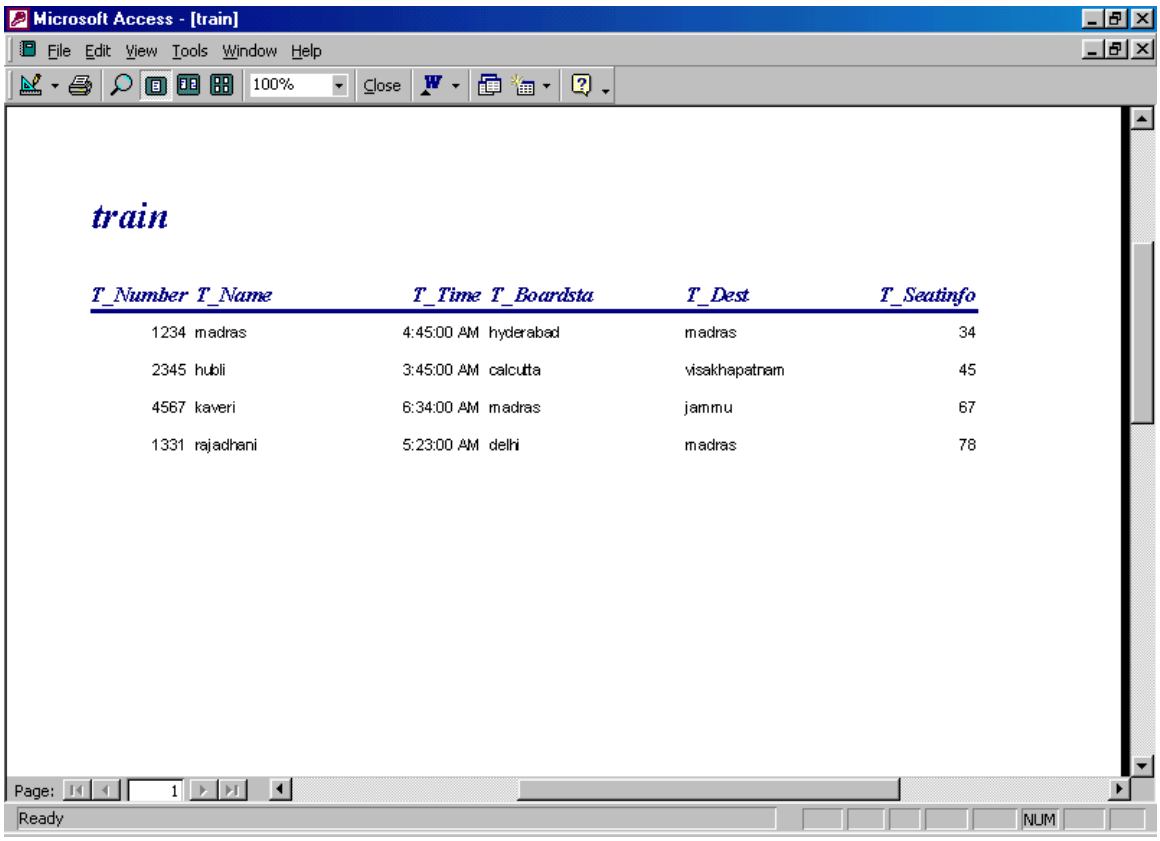


This is the form that is to modified the train details. This form consists of no of seats textbox, which shows the remaining seats present in the train. The no of seats value is automatically changed when we create or cancel reservation. This is the form which consist of two buttons for the purpose of create reservation and to cancel reservation. When the seats text box has value zero than its, says no seats are remaining in the train to create reservation.

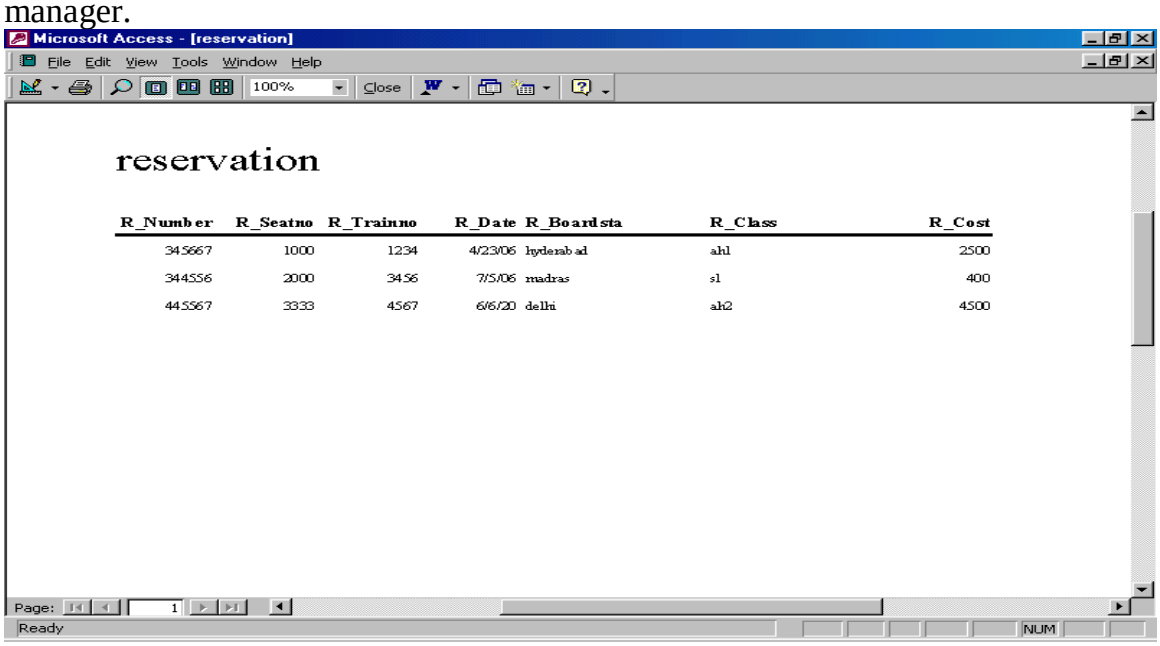


Reports for Railway Reservation System

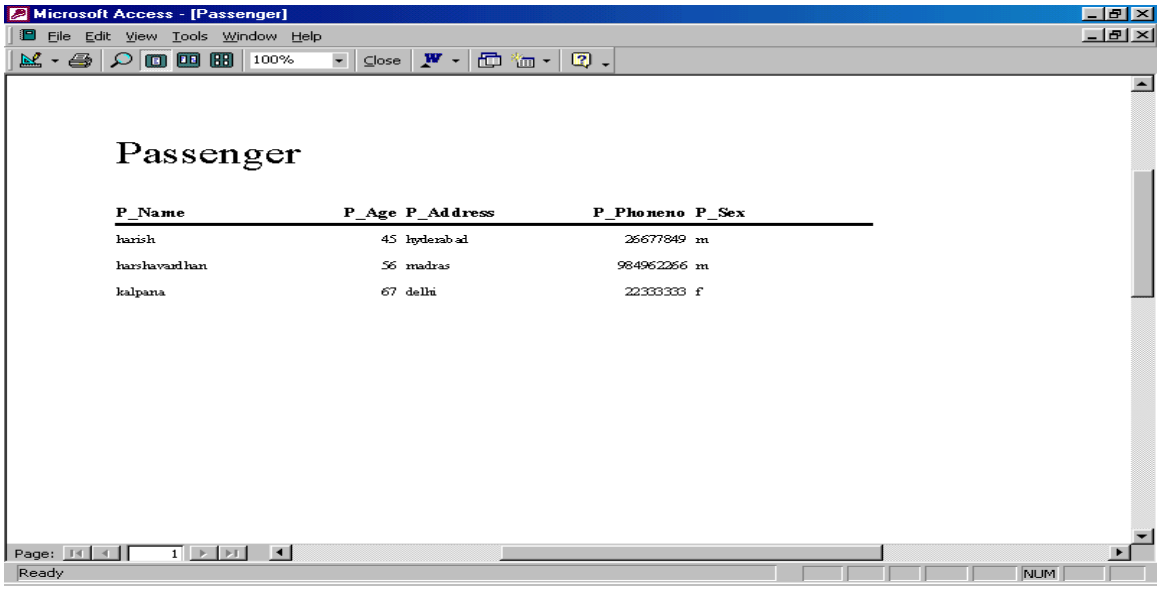
This is report generated by manager after modifying train details.



This is report generated by the clerk in order to give the reservation details to the manager.



This is report generated by the clerk about the passenger details who take the reservation.



Session – 7

This is the program, which takes two matrices as input and generates multiplication of two matrices as output. In order to multiply the two matrices there is a condition for the two matrices, is given below.

- The no. of columns of the first matrix is equal to the no. of rows of the second matrix, then only it is possible to multiply two matrices, otherwise it is not possible to multiply two matrices.

Program

```
#include<stdio.h>
#include<conio.h>
void input(int **a,int r1, int c1)
{
int i,j;
printf("\n enter 1st matrix elements  \n");
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
else
{
for(i=0;i<r1;i++)
```

```
for(j=0;j<c1;j++)
scanf("%d",&*(*(a+i)+j));
}
}
void output(int **a, int r2, int c2)
{
int i,j;
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
else
{
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
printf("%d\t",*(*(a+i)+j));
printf("\n");
} }
}

void multiply(int **a, int **b, int r1, int c1, int r2, int c2)
{
int i,j,k,c[10][10];
for(i=0;i<r1;i++)
for(k=0;k<c2;k++)
{
c[i][k]=0;
for(j=0;j<c1;j++)
*(*(c+i)+k)= *(*(c+i)+k)+ *(*(a+i)+j)* *(*(b+j)+k);
}
printf("\n the multiplication of matrices is \n");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
printf("%d\t",*(*(c+i)+j));
printf("\n");
}
}
main()
{
int r1,c1,r2,c2,**a, **b, **c;
clrscr();
printf("\n enter the number of rows,columns for matrix1\n");
scanf("%d%d",&r1,&c1);
```

```
printf("\n enter the number of rows,columns for matrix2\n");
scanf("%d%d",&r2,&c2);
if(c1!=r2)
{
printf("\n unable to multiply");
return;
}
else
{
input(a,r1,c1);
input(b,r2,c2);
multiply(a,b,r1,c1,r2,c2);
output(a,r1,c1);
output(b,r2,c2);

}
getch();
}
```

Output:

```
 enter the number of rows, columns for matrix1
3
3

 enter the number of rows, columns for matrix2
3
3

 enter 1st matrix elements
1
2
3
4
5
6
7
8
9

 enter 2nd matrix elements
1
1
1
1
1
```

```
1
1
1
1

 the first matrix is
1       2       3
4       5       6
7       8       9

 the second matrix is
1       1       1
1       1       1
1       1       1

 the multiplication of matrices is
6       6       6
15      15      15
24      24      24
```

Session – 8

Here we want to generate test cases that will completely test the program given above. This is the program, which takes two matrices as input and generates multiplication of two matrices as output. In order to multiply the two matrices there is a condition for the two matrices, is given below.

- The no. of columns of the first matrix is equal to the no. of rows of the second matrix, then only it is possible to multiply two matrices, otherwise it is not possible to multiply two matrices.

Unit Testing

Unit testing is procedure used to verify particular segment of source code is working properly. The main idea about this testing is to generate the test cases for all function or methods. The main goal of unit testing is isolate each part of program and show individual parts are correct. In above program, there are three functions, which perform input, output, and multiplication of two matrices. Let us consider the first function.

In main function let us consider

```
if(c1!=r2)
{
printf("\n unable to multiply");
return;
}
else
```

```
{
input(a,r1,c1);
input(b,r2,c2);
multiply(a,b,r1,c1,r2,c2);
output(a,r1,c1);
output(b,r2,c2);

}
```

The above source code determines the no of columns of the first matrix equal to the no. of rows of the second matrix then only the multiplication is possible.

Test Case	c1	r2	Expected Output
1	3	3	Multiplication of two matrices is possible
2	3	4	Multiplication of two matrices is impossible

This is a segment of input function.

```
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
else
{
for(i=0;i<r1;i++)
for(j=0;j<c1;j++)
scanf("%d",&a[i][j]);
}
```

The above code reprgents the no. of rows and no. of columns that are to be given is less than the size of matrix. If the it is not possible to enter the value in the matrix is not greater than its size.

Test Case	r1	c1	Expected Output
1	3	3	Entering of matrix
2	10	10	Unable to enter matrix
3	11	11	Unable to enter matrix

```
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
else
```

```
{
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
printf("%d\t",a[i][j]);
printf("\n");
}
}
```

The above code is a segment of output function. This code represent the no .of rows and no. of columns of the matrix is less than the size of the matrix if we gave the no. of rows and no. of columns greater than or equal to size of the array, then we get unexpected values as output

Test Case	r1	c1	Expected Output
1	3	3	Shows output of matrix
2	10	10	Unexpected values
3	11	11	Unexpected values

Module Testing

Module testing is procedure used to verify the source code is working properly or not. The main idea about this testing is to generate the test cases for all function or methods. In above program, there are three functions, which perform input, output, and multiplication of two matrices. Let us consider the first function.

Input Function

```
void input(int a[10][10],int r1, int c1)
{
int i,j;
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
printf("\n enter 1st matrix elements  \n");
for(i=0;i<r1;i++)
for(j=0;j<c1;j++)
scanf("%d",&a[i][j]);
}
```

In above function there are three inputs. The inputs are no. of rows and no. of columns. The no. of rows and no. of columns can not be greater than the size of the array declared. If the no. of rows and no. of columns is greater than are equal to the size of the array, than it is not possible to enter the values into the array.

Test Case	r1	c1	Expected Output
1	3	3	Entering of matrix
2	10	10	Unable to enter matrix
3	11	11	Unable to enter matrix

The second function in the above program is given below

Output Function

```
void output(int a[10][10], int r2, int c2)
{
int i,j;
if (r1>=10||c1>=10)
{
printf("\n Unable to enter matrix");
return;
}
else
{
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
printf("%d\t",a[i][j]);
printf("\n");
} }
}
```

In above function there are three inputs. In this function, the no. of rows and no. of columns that are to be entered is not greater than size of the array.
If the no. of rows and no. of columns is less than the size of the array then it shows that output of the values in the array. If the no. of rows and no. of columns is greater than are equal to the size of the array then the function shows unexpected values because of more than the size of array.

Test Case	r1	c1	Expected Output
1	3	3	Shows output of matrix
2	10	10	Unexpected values
3	11	11	Unexpected values

The next function of above program is *Multiplication Function*

```
void multiply(int a[10][10], b[10][10], int r1, int c1, int r2, int c2)
{
int i,j,k,c[10][10];
for(i=0;i<r1;i++)
for(k=0;k<c2;k++)
{
c[i][k]=0;
for(j=0;j<c1;j++)
c[i][k]=c[i][k]+a[i][j]*b[j][k];
}
printf("\n the multiplication of matrices is \n");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
printf("%d\t",c[i][j]);
printf("\n");
}
}
```

In above function there are six inputs.

- If the no. of rows and no. of columns of the two matrices less than the size of the respected matrix and the column of the first matrix is equal to the row of the second matrix then we get the correct multiplication of two matrices.
- If the no. of rows and no. of columns is greater than or equal to the size of the respected array and column of the first matrix is equal to the row of the second matrix, then multiplication is possible but we get the unexpected values because of the more than the size of the array.
- If the no. of rows and no. of columns is less than the size of the respected array and column of the first matrix is not equal to the row of the second matrix, then matrix multiplication is not possible.

Test Case	r1	c1	r2	c2	Expected Output
1	3	3	4	4	Unable to multiply
2	3	3	3	3	Multiplication is possible
3	3	3	10	10	Unable to multiply
4	3	3	11	11	Unable to multiply
5	10	10	3	3	Unable to multiply
6	11	11	3	3	Unable to multiply
7	10	10	10	10	Unable to multiply
8	11	11	11	11	Unable to multiply

Integrated Testing

Integration testing is the phase of software testing in which individual software modules are combined and tested as a group. Integrating testing as takes as its input, modules that have been checked out by unit testing groups them in larger aggregates, applies this test in order to know whether the system generate the actual output or not.

The main aim of this program is to calculate multiplication of two matrices. The multiplication function takes two matrices as input and generates the multiplication as matrix. We must determine the two matrices before the multiplication of two matrices then only we get the multiplication of two matrices. If we don't determine any of two matrices then we get unexpected values as multiplication of two matrices by taking the default values stored in matrix.

Test Case	Matrix a	Matrix b	Expected Output
1	determined	determined	Expected output as multiplication of two matrices
2	undetermined	determined	Unexpected output as multiplication of two matrices
3	determined	undetermined	Unexpected output as multiplication of two matrices
4	undetermined	Undetermined	Unexpected output as multiplication of two matrices`

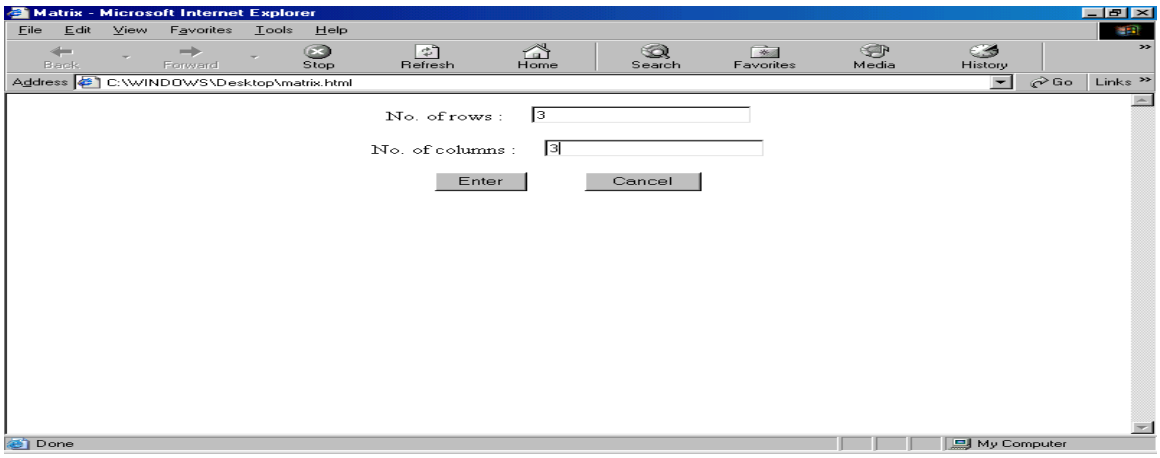
The main aim of the above program is not achieved if we undetermined any of the two matrices which are taken as input for multiplication of two matrices.

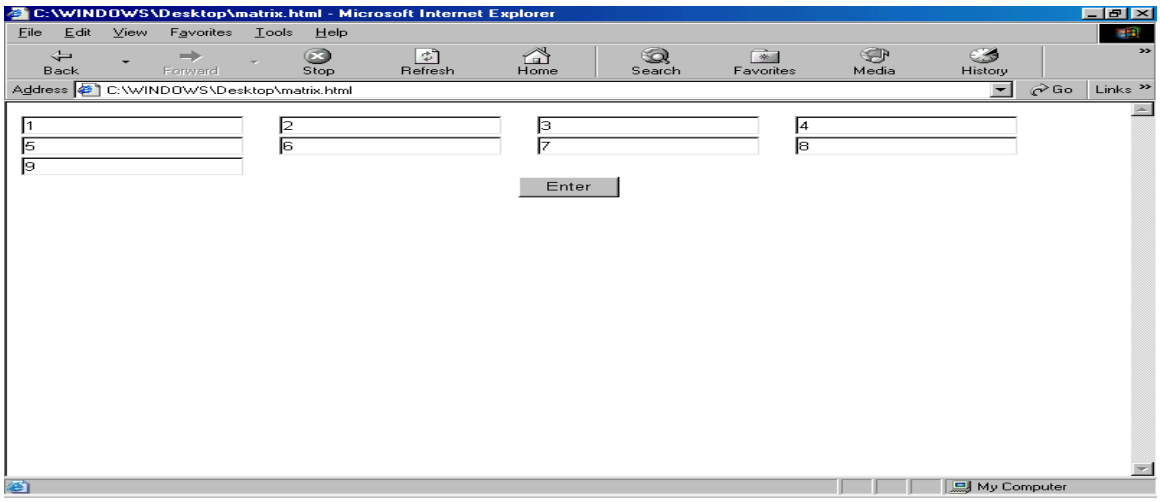
Session – 9

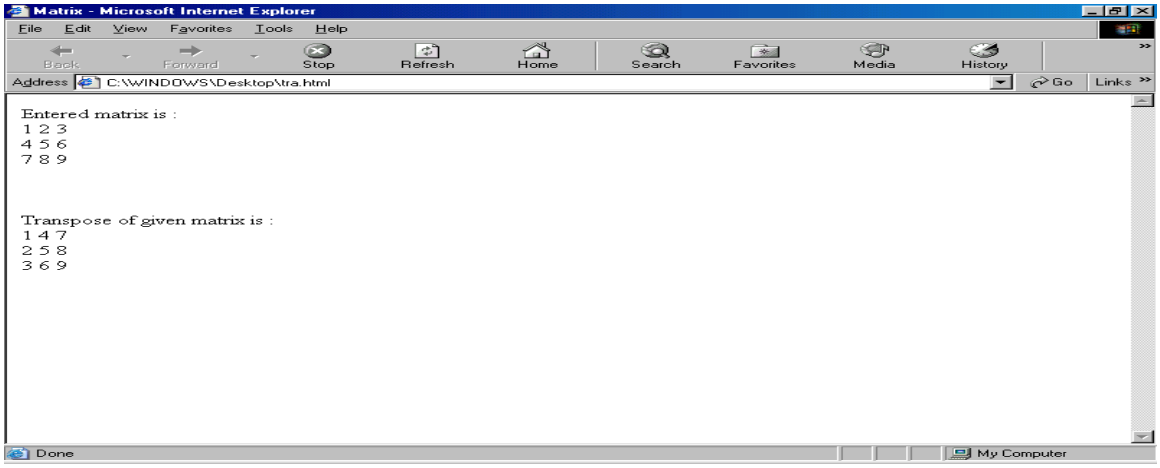
This is web page, which accepts Matrix as input and generates the transpose of the given matrix. This web page consists of two text boxes and two buttons. In two text boxes we must enter the no of rows and no of columns and the web page generates the no of text boxes for entering the values of the matrix depends up on the number of rows and columns given. In that textboxes we must enter values and then the web page generates the transpose of the given matrix.

```
<html><head>
<title>Matrix</title>
<script language="javascript">
function enter()
{
var a=form1.text1.value;
var b=form1.text2.value;
if(a==0||b==0)
alert(" you must enter values in textbox");
else
{
document.writeln("<form name='form2'>");
for(i=0;i<a;i++)
for(j=0;j<b;j++)
document.writeln("<input type=text
name='text[i][j]'>     ");
document.writeln("<br><center><input type=button value=' Enter '
onClick='transpose()'></center>");
document.writeln("</form>");
}
}
function transpose()
{
for(i=0;i<a;i++)
for(j=0;j<b;j++)
a[i][j]=form2.text[i][j];
document.writeln("Entered matrix is :");
for(i=0;i<a;i++)
{
for(j=0;j<b;j++)
document.writeln(a[i][j]+"\t");
document.writeln("<br>");
}
document.writeln(<BR><BR><BR>");
document.writeln("Transpose of given matrix is :");
for(i=0;i<a;i++)
for(j=0;j<b;j++)
```

```
b[i][j]=a[j][i];
for(i=0;i<a;i++)
{
for(j=0;j<b;j++)
document.writeln(b[i][j]+"\t");
document.writeln("<br>");
}
}
</script></head>
<body><center>
<form name="form1">
No. of rows :     <input type=text name="text1"><br><br>
No. of columns :     <input type=text name="text2"><br><br>
<input type=button value=" Enter " onClick='enter()'>         
<input type=button value=" Cancel ">
</form></center></body></html>
```







Session – 10

We have to generate the test cases for the above web page. Normally there are two types of testing. They are

i) White Box Testing
ii) Black Box Testing

Black Box Testing

In black box testing can help to get the design and coding correct with respect to the specification. Black box testing is mainly used to test the functionality and features of the system. In black box testing, there are two strategies followed as given below.

1) Boundary Value Analysis:

In the web page there are two text boxes, which has to enter no. of rows and no. of columns. We must enter both no. of columns and no. of rows for matrix. If we don't given any of the two values then we don't get the matrix.

Test Case	r1	c1	Expected Output
1	-	-	We get an error message
2	10	-	We get an error message
3	-	11	We get an error message
4	10	11	We get text boxes to enter matrix values

In order to complete our aim of calculating the transpose of a matrix, we have to enter both no. of rows and no. of columns for the matrix.

Test Case	Input	Expected Output
1	Clicking Enter Button	Web page having text boxes
2	Clicking Cancel Button	We get message

On successfully entering the no. of rows and no. of columns, when we click the Enter button we get the web page having the text boxes to enter the value of the matrix. When we click the Cancel button then we get a message.

If we enter both no. of rows and no. of columns then we get a web page of have text boxes of number equal to the no . of rows * no. of columns. In order to get the text boxes to enter values of the matrix we have to enter the both the no. of rows and no. of columns.

We have to enter the values in all the text boxes, then only the values are entered into matrix. Suppose we gave 2 * 2 matrix then we get 4 text boxes.

Test Case	**T1**	**T2**	**T3**	**T4**	**Expected Output**
1	-	-	-	-	Error Message
2	3	-	-	-	Error Message
3	-	4	-	-	Error Message
4	-	-	7	-	Error Message
5	-	-	-	10	Error Message
6	11	10	-	-	Error Message
7	14	13	55	-	Error Message
8	-	12	11	10	Error Message
9	12	11	-	14	Error Message
10	14	-	15	16	Error Message
11	-	-	13	12	Error Message
12	10	11	12	13	Transpose Matrix

When we don’t enter the value into any text box, then we get a error message. On successfully entering into all text boxes we get the transpose of the give matrix, in the next web page