

# **Railway Management System**

**Name:** Purna Srivalli Koravi

**Semester:** Fall

**Course:** Database Management Systems

**Instructor name:** Dr. Suhair H Amer

## **HARDWARE AND SOFTWARE COMPONENTS**

### **HARDWARE COMPONENTS**

**RAM** : 412 MB or more

**Hard disk** : 10 GB or more

### **SOFTWARE COMPONENTS**

**Operating System** : Windows 10

**Front end** : NetBeans IDE 8.2

**Back end** : My SQL Workbench 8.0 CE

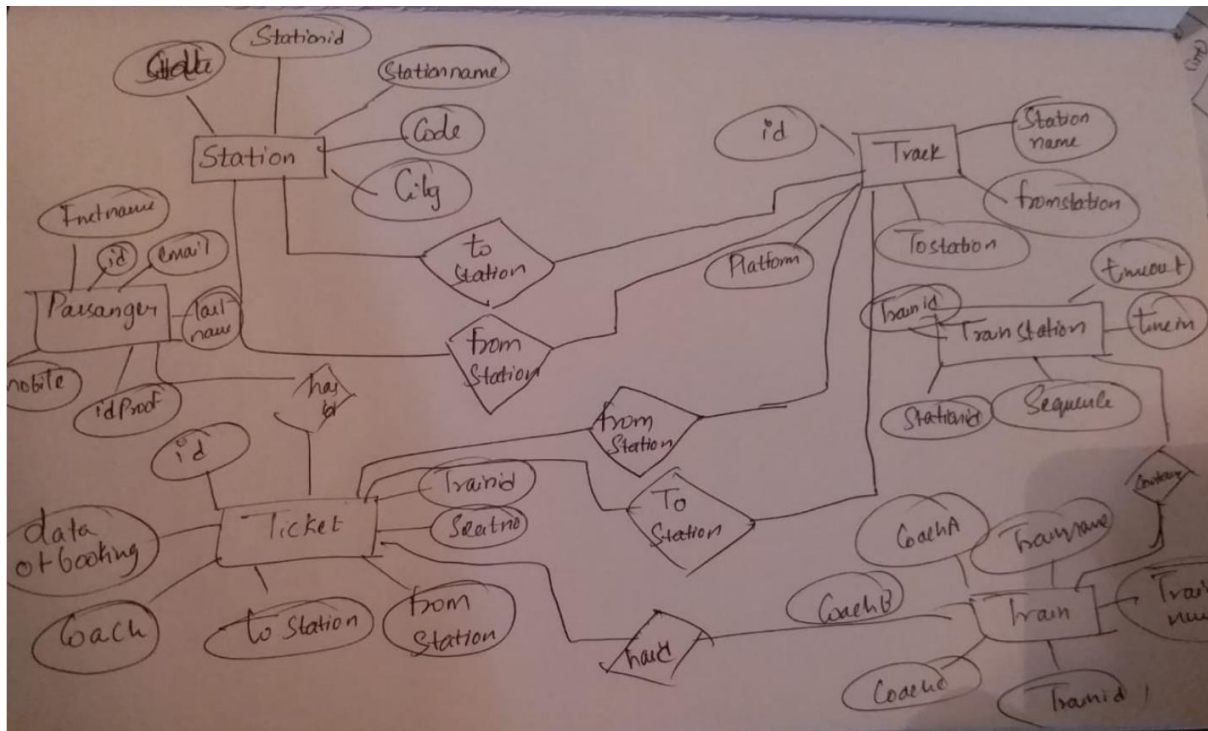
## **DESCRIPTION**

Application built to handle few operations such as displaying the train schedule and Enabling the user to book tickets for his/her journey from desired locations. Tools used are MySQL Work bench, Visual Studio code IDE. Languages used are MySQL.

The below tables are created to suit the requirements of searching the trains and booking a ticket. The track table is created in relation with stations with a unique constrain on both The train table is created with 3 coaches which will be used to store the availability of seats in that coach for that train.

For maintaining the train schedule trainsstations is used along with the sequence for the order of the stations in the route.

# ER DIAGRAM:



## TABLES CREATION:

trackstationtracktrack - tabletrackstationtracktraintrainstationtraintrainstationpasseneypassene - tablepassene - table

Table Name:

Schema: **railwaymanagement**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<input type="checkbox"/> id	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> firstname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> lastname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> email	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> mobile	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> proof	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:

Charset/Collation:

Comments:

Data Type:

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

ColumnsIndexesForeign KeysTriggersPartitioningOptions

Table Name:

Schema: **railwaymanagement**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<input type="checkbox"/> Stationid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> stationname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> code	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> state	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<input type="checkbox"/> city	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:

Charset/Collation:

Comments:

Data Type:

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

ColumnsIndexesForeign KeysTriggersPartitioningOptions

Apply

Revert

track - table

track

station

track

train

trainstation

train

trainstation

passenegr

passenegr - table

passenegr - table

station - table

ticket - table

SQLAddit

Table Name:

Schema: **railwaymanagement**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
passengerid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
fromstation	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tostation	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
seatnumber	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
coach	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
dateofbooking	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
journeydates	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:

Charset/Collation:

Comments:

Data Type:

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns

Indexes

Foreign Keys

Triggers

Partitioning

Options

Apply

Revert

Context

Output

Action Output

#

Time

Action

Message

Table Name:

Schema: **railwaymanagement**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
trackid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
fromstationid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tostationid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
platform	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
stationname	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:

Charset/Collation:

Comments:

Data Type:

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ U

☐ Binary ☐ Unsigned ☐ Z

☐ Auto Increment ☐ Generated

Columns

Indexes

Foreign Keys

Triggers

Partitioning

Options

Table Name:

Schema: **railwaymanagement**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
trainid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
trainname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
trainnumber	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
coachA	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
coachB	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
coachC	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:

Charset/Collation:

Comments:

Data Type:

Expression:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☒ Generated

Columns

Indexes

Foreign Keys

Triggers

Partitioning

Options

Apply

Revert




Table Name:

Schema: **railwaymanagement**

Charset/Collation: 






utf8mb4

utf8mb4\_0900\_ai\_ci

Engine: 

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 trainid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 stationid	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 timein	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 timeout	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 sequencenumber	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:

Charset/Collation: 

Default Charset

Default Collation

Comments:

Data Type:

Expression:

Storage: 

☐ Virtual

☐ Stored

☐ Primary Key

☐ Not Null

☐

☐ Binary

☐ Unsigned

☐

☐ Auto Increment

☒ Generated

Columns

Indexes

Foreign Keys

Triggers

Partitioning

Options

## Insertion of data:

s

```
9 ▾ insert into railwaymanagement.station values(995, mexicoexp , MU , MEXICO , mexi );
10 • Insert into railwaymanagement.station values(189,'torontoexp','tO','toronto','canada');
```

e

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Stationid	stationname	code	state	city
124	Missouriexp	MSO	Missouri	cape
224	texasexp	tO	texas	girardeau
144	Carolinaexp	cO	caroline	caro
154	virginiaexp	vO	virginia	virginiaa
8764	dallasexp	dO	dallas	cdal
678	floridaexp	fO	florida	flor
600	califonia	coO	california	cal
993	mexicoexp	MO	Mexico	mexi
189	torontoexp	tO	toronto	canada
198	orlandoexp	onO	orlando	orla

station 3 x

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

trainid	trainname	trainnumber	coachA	coachB	coachC
444	hogwarts	94	2	4	1
456	cape	44	3	4	2
984	missouri	78	1	4	5
444	hogwarts	94	5	4	1
444	carolina	94	2	4	1
444	mexico	54	2	4	3
444	missouri	84	2	8	1
444	chicago	84	2	4	4
444	virginia	94	2	9	1
674	california	84	8	4	1

train 2 train 3 x

Output

Action Output

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

trackid	fromstationid	tostationid	platform	stationname
946	3	1	2	missouri
946	3	1	2	newyork
946	3	5	9	cape
976	7	1	2	mexico
846	3	1	2	texas
966	3	1	2	dallas
546	3	7	5	carolina
846	3	1	2	colombus
916	3	2	6	virginia
546	6	1	2	chicago

rack 2 track 3 x

Output



Result Grid						Filter Rows:	Export:	Wrap Cell Content:
trainid	stationid	timein	timeout	sequencenumber				
123	247	02:00:00	04:00:00	3				
123	647	09:00:00	04:00:00	1				
823	247	02:00:00	04:00:00	2				
123	247	02:00:00	09:00:00	3				
123	547	09:00:00	04:00:00	5				
523	247	02:00:00	04:00:00	3				
123	847	02:00:00	08:00:00	4				
193	247	02:00:00	04:00:00	3				
123	247	09:00:00	04:00:00	4				
523	747	08:00:00	09:00:00	3				

trainstation 2    trainstation 3 x

Output

Result Grid						Filter Rows:	Export:	Wrap Cell Content:
id	firstname	lastname	email	mobile	proof			
10	keety	sdh	chand8u@gmail.com	984989898	184567			
6787	siri	koravi	sityy@gmail.com	999999	license			
147	svalli	koravi	siri@gmail.com	99645349	license			
677	aag	koravi	siddw@gmail.com	999999	license			
147	ghj	koravi	siri@gmail.com	9967899	license			
1679	john	koravi	sffg@gmail.com	9967899	license			
156	rose	koravi	siri@gmail.com	9964399	license			
147	siri	koravi	ster@gmail.com	945999	license			
697	bryce	koravi	sddf@gmail.com	9567999	license			
147	siri	koravi	firi@gmail.com	9890999	license			
147	clav	koravi	edadi@gmail.com	3456000	license			

passeneger 2    passeneger 3 x

Output

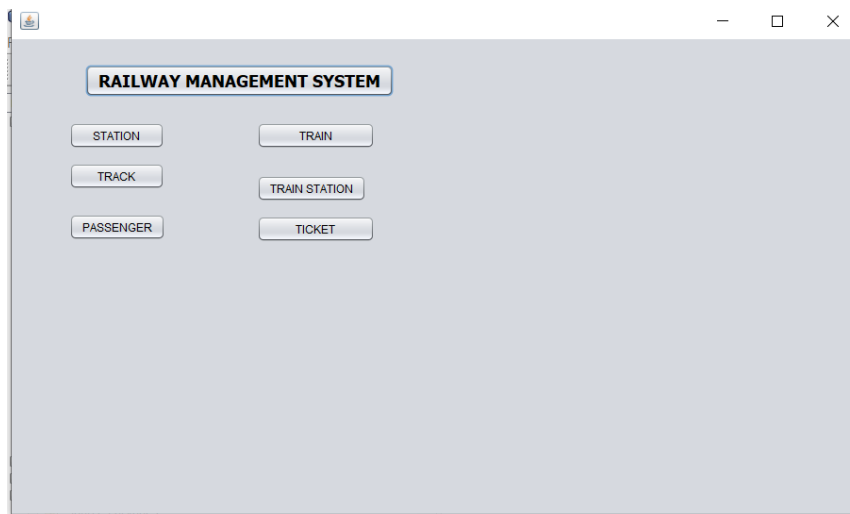
Action Output

id	passengerid	fromstation	tostation	seatnumber	coach	dateofbooking	journeydates	trainid
123456	1234	3333	2546	43	A	2019-12-09 20:16:33	2019-12-08 00:00:00	1234
123445	1244	3343	2146	43	B	2019-12-09 20:16:33	2019-12-10 20:16:33	1235
123457	1254	3353	2246	43	A	2019-12-19 20:16:34	2019-12-20 20:16:36	1236
123458	1264	3363	2346	43	A	2019-12-29 20:16:35	2019-12-30 20:16:30	1237
123459	1274	3373	2446	43	B	2019-12-15 20:16:36	2019-12-16 20:16:31	1238
123466	1284	3383	2646	43	A	2019-12-16 20:16:37	2019-12-17 20:16:38	1239
123476	1294	3393	2746	43	B	2019-12-17 20:16:38	2019-12-18 20:16:33	1230
123486	1204	3343	2846	43	A	2019-12-09 20:16:39	2019-12-10 20:16:33	1231
123496	1214	3303	2946	43	A	2019-12-18 20:16:40	2019-12-19 20:16:33	1232
123406	1224	3323	2046	43	C	2019-12-19 20:16:41	2019-12-20 20:16:33	1233

ticket 3 x

Output

## GUI INTERFACE:



## LABELS:

### Copy of code:

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package javaapplication6;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import javax.swing.JOptionPane;  
  
/**  
 *
```

```

* @author Srivalli
*/

public class trainstation extends javax.swing.JFrame {

    /**
     * Creates new form trainstation
     */
    public trainstation() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        trainid = new javax.swing.JTextField();
        stationid = new javax.swing.JTextField();
        timein = new javax.swing.JTextField();

```

```
timeout = new javax.swing.JTextField();
sequence = new javax.swing.JTextField();
submit = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel1.setText("Train Station");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(294,
38, -1, -1));

jLabel2.setText("Train Id");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(133,
105, -1, -1));

jLabel3.setText("Station Id");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(133,
156, -1, -1));

jLabel4.setText("Time In");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(133,
210, -1, -1));

jLabel5.setText("Time Out");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(133,
271, -1, -1));

jLabel6.setText("Sequence");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(133,
334, -1, -1));
```

```
getContentPane().add(trainid, new org.netbeans.lib.awtextra.AbsoluteConstraints(511, 105, 69, -1));
```

```
getContentPane().add(stationid, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(511, 164, 69, -1));
```

```
getContentPane().add(timein, new org.netbeans.lib.awtextra.AbsoluteConstraints(511, 210, 69, -1));
```

```
getContentPane().add(timeout, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(511, 268, 69, -1));
```

```
getContentPane().add(sequence, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(511, 331, 69, -1));
```

```
submit.setText("Submit");
```

```
submit.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        submitActionPerformed(evt);  
    }  
});
```

```
getContentPane().add(submit, new org.netbeans.lib.awtextra.AbsoluteConstraints(294, 392, -1, -1));
```

```
pack();
```

```
}// </editor-fold>
```

```
private void submitActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    int TrainId=Integer.parseInt(trainid.getText());
```

```
    int stationId=Integer.parseInt(stationid.getText());
```

```
    String Timein = timein.getText();
```

```
    String Timeout = timeout.getText();
```

```
    int Sequence=Integer.parseInt(sequence.getText());
```

```

Connection conn=null;

PreparedStatement pstmt=null;

try{
    Class.forName("com.mysql.jdbc.Driver");
    //demo is the name of the database
    conn= DriverManager.getConnection("jdbc:mysql:///javaapplication6", "root",
    "Sriva11i");//replace XXXXXX with you password

    //employees is the name of the table
    //4 questions marks for 4 fields in the table that i want to insert into
    pstmt=conn.prepareStatement("insert into Station values(?,?,?,?)");

    pstmt.TrainId(1,trainid);
    pstmt.stationId(2, stationid);
    pstmt.Timein(3, timein);
    pstmt.Timeout(4, timeout);
    pstmt.Sequence(4, sequence);

    int i=pstmt.executeUpdate();//will update database
    if (i>0) {
        JOptionPane.showMessageDialog(null, "Data is saved");
    }
else {

        JOptionPane.showMessageDialog(null, "Data not saved");
    }
}

catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}

```

```

    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
    feel.
        * For details see
        http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(trainstation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(trainstation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

```

```
java.util.logging.Logger.getLogger(trainstation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(trainstation.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new trainstation().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JLabel jLabel5;
```

```
private javax.swing.JLabel jLabel6;
```

```
private javax.swing.JTextField sequence;
```

```
private javax.swing.JTextField stationid;
```

```
private javax.swing.JButton submit;
```

```
private javax.swing.JTextField timein;
```

```
private javax.swing.JTextField timeout;
```

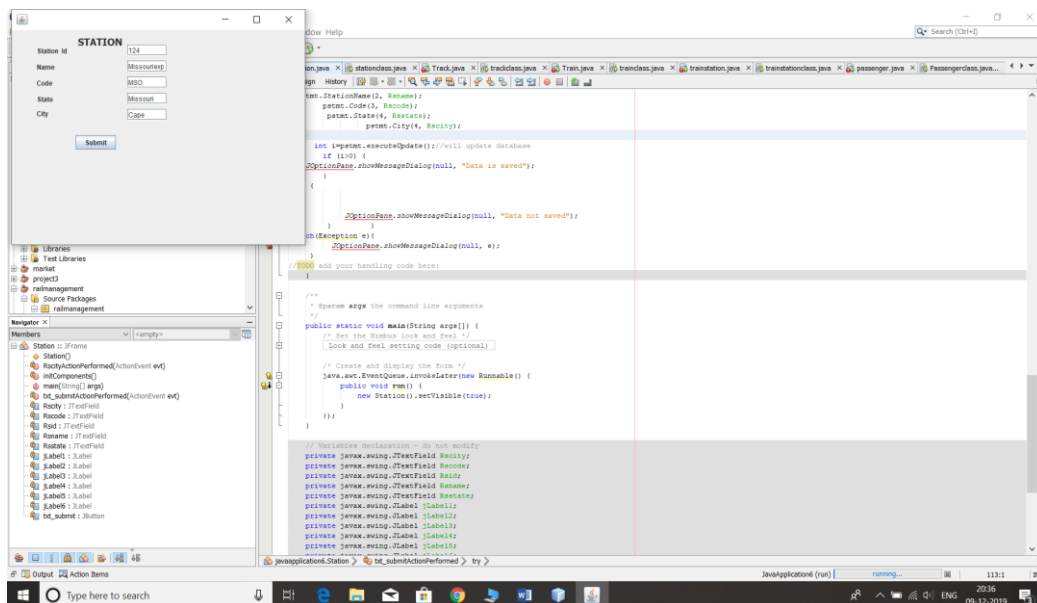
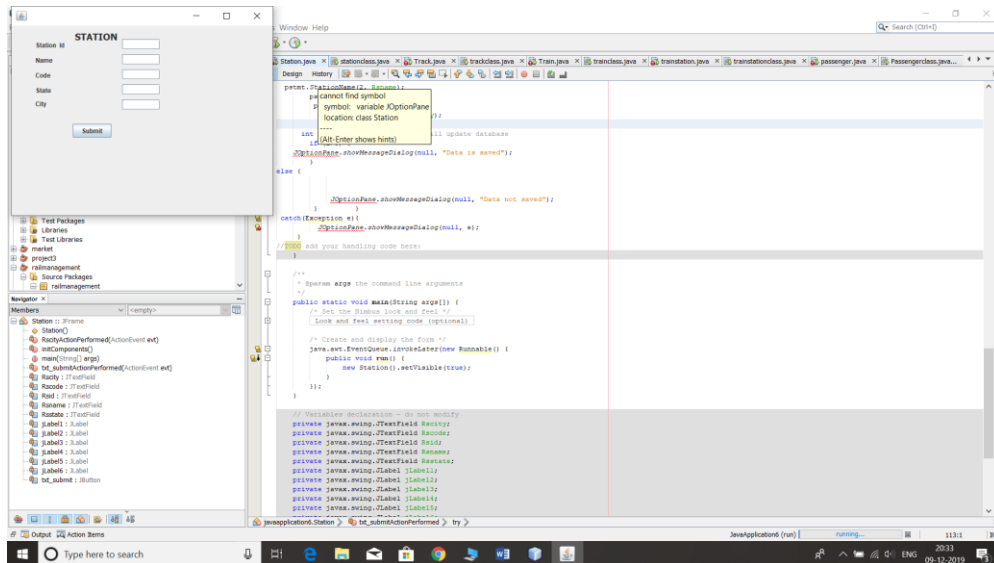
```
private javax.swing.JTextField trainid;
```



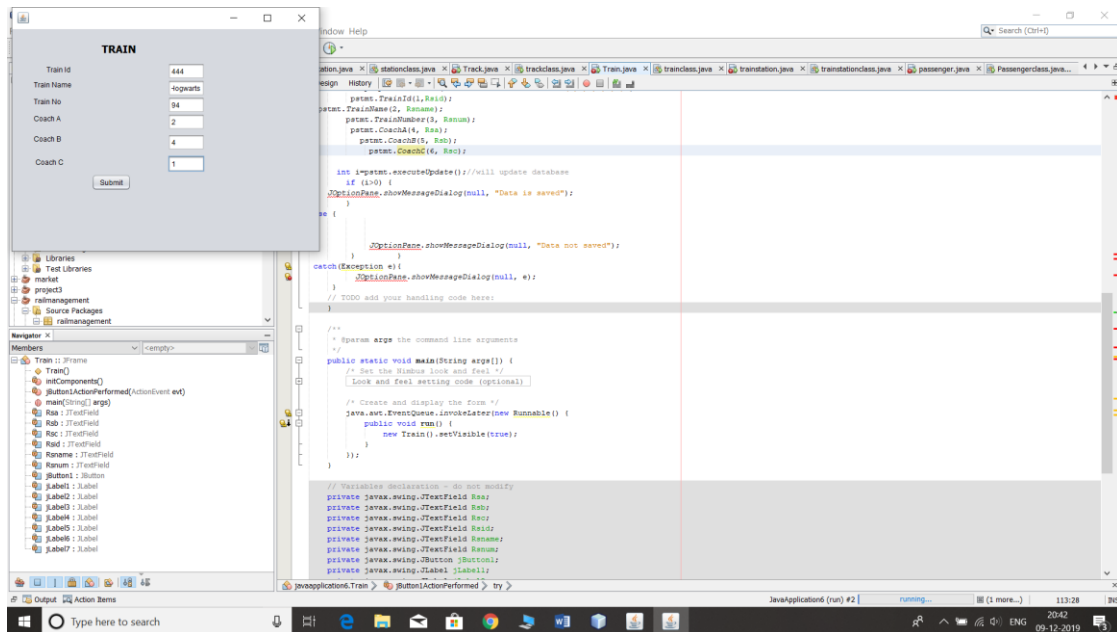
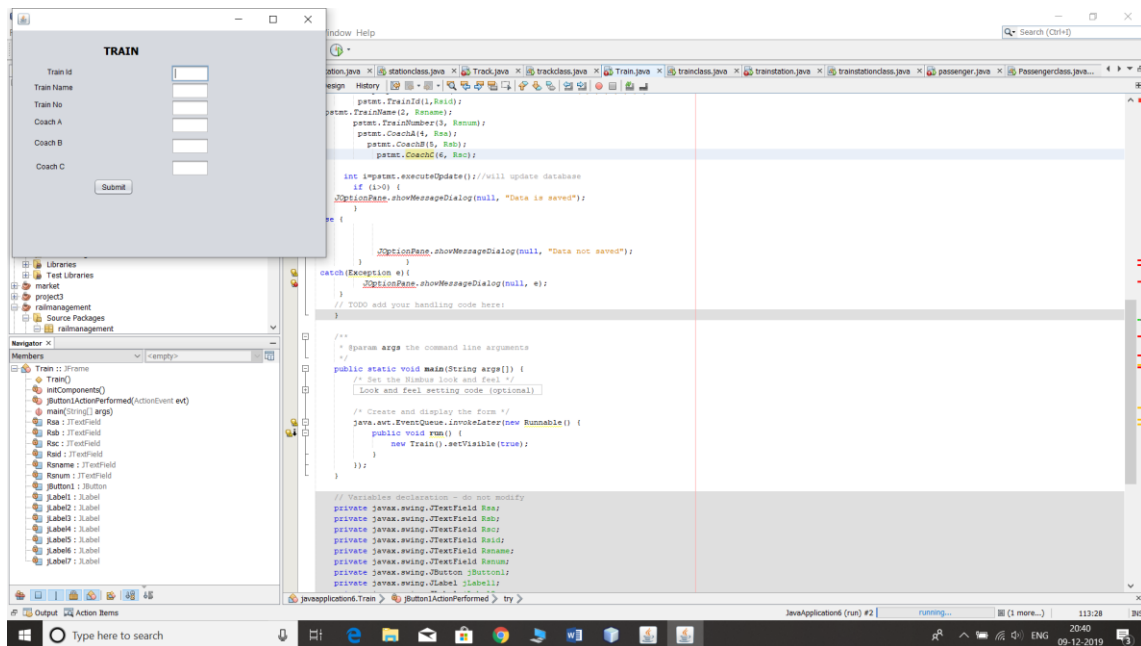
// End of variables declaration

}

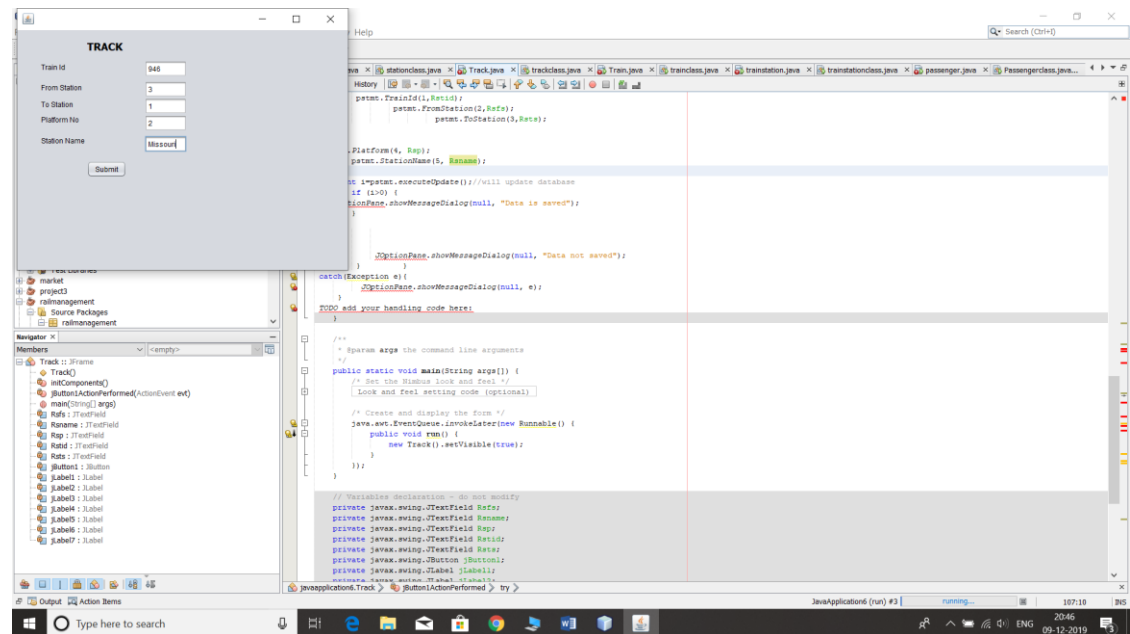
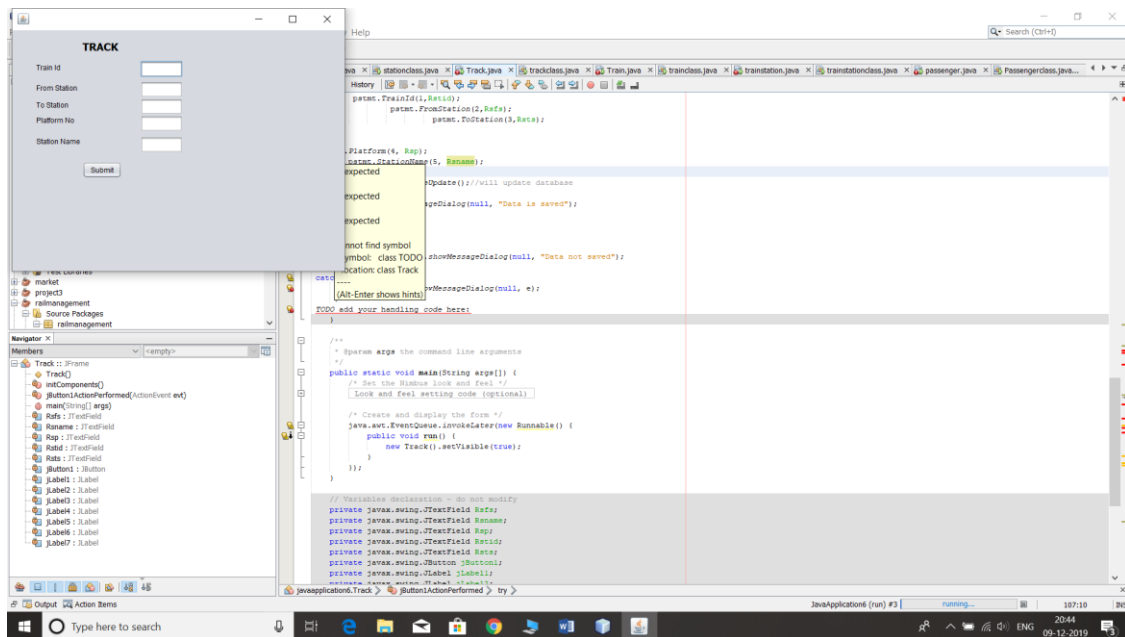
Station:



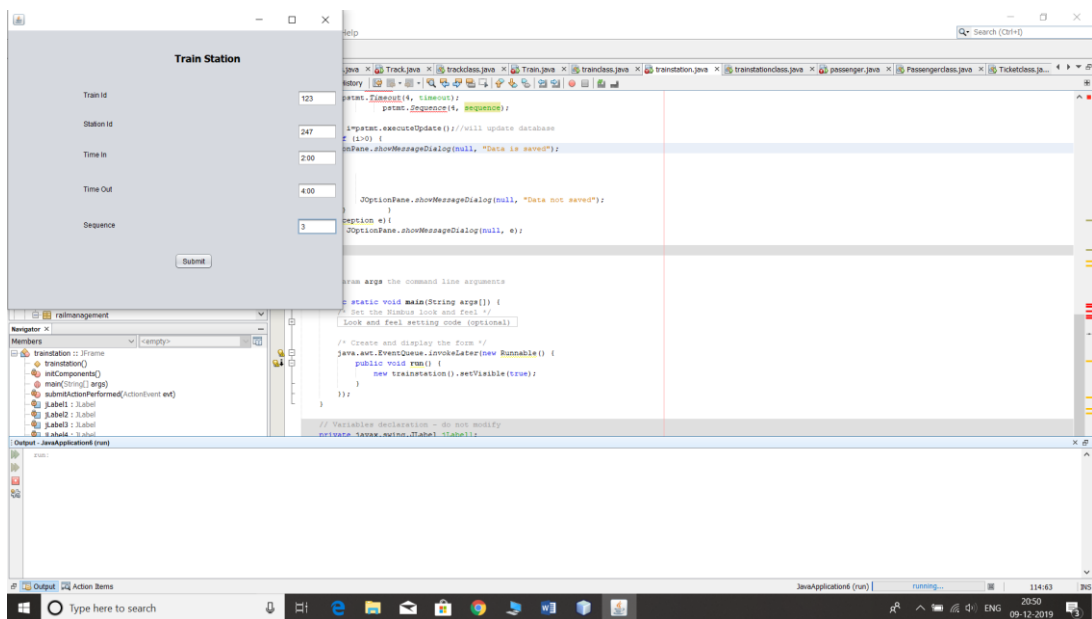
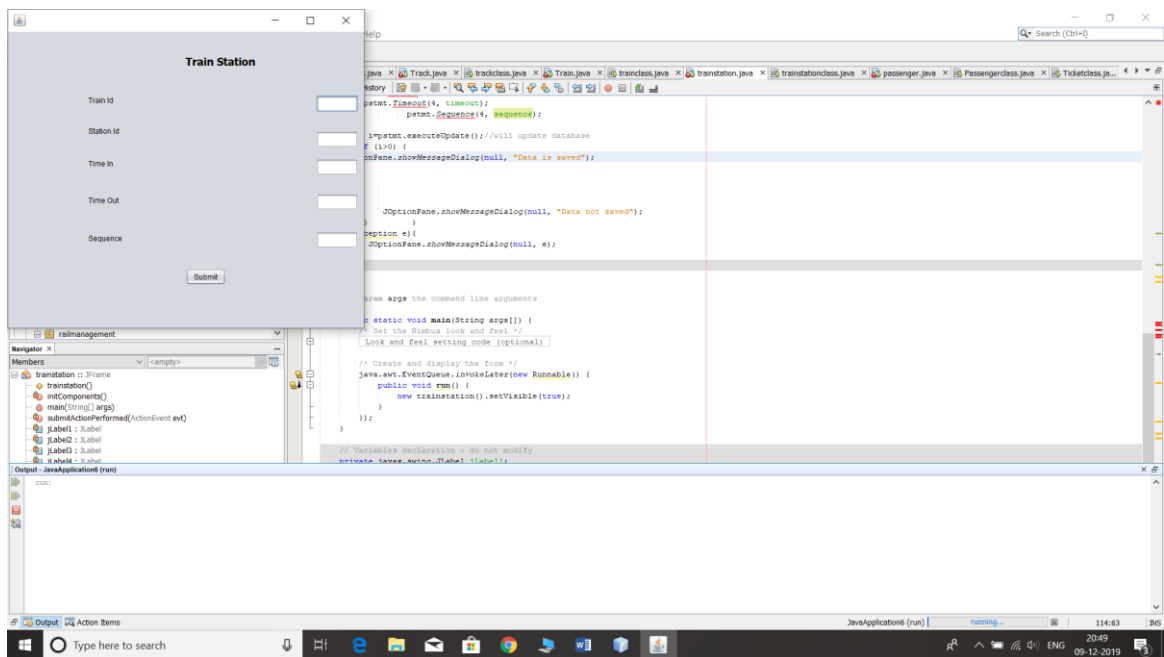
## Train:



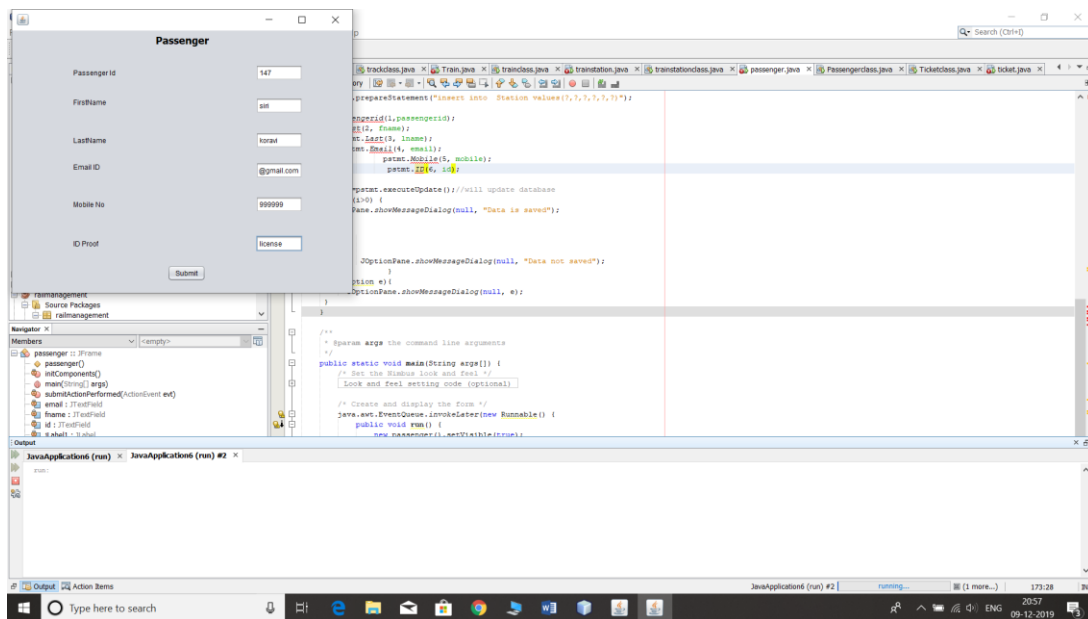
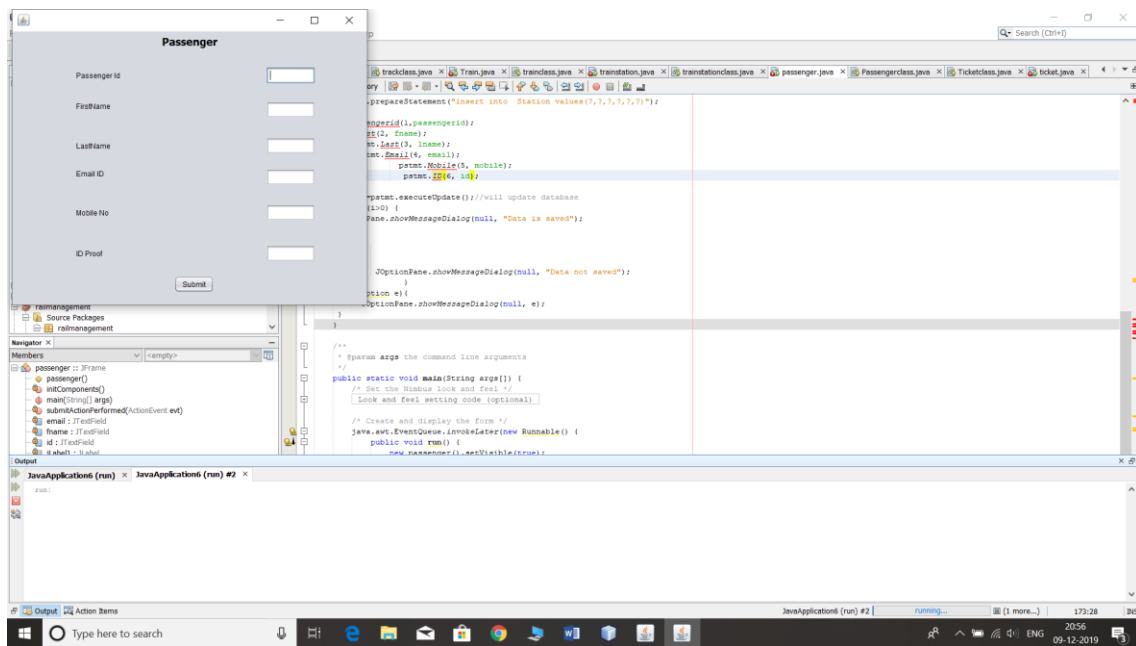
## Track:



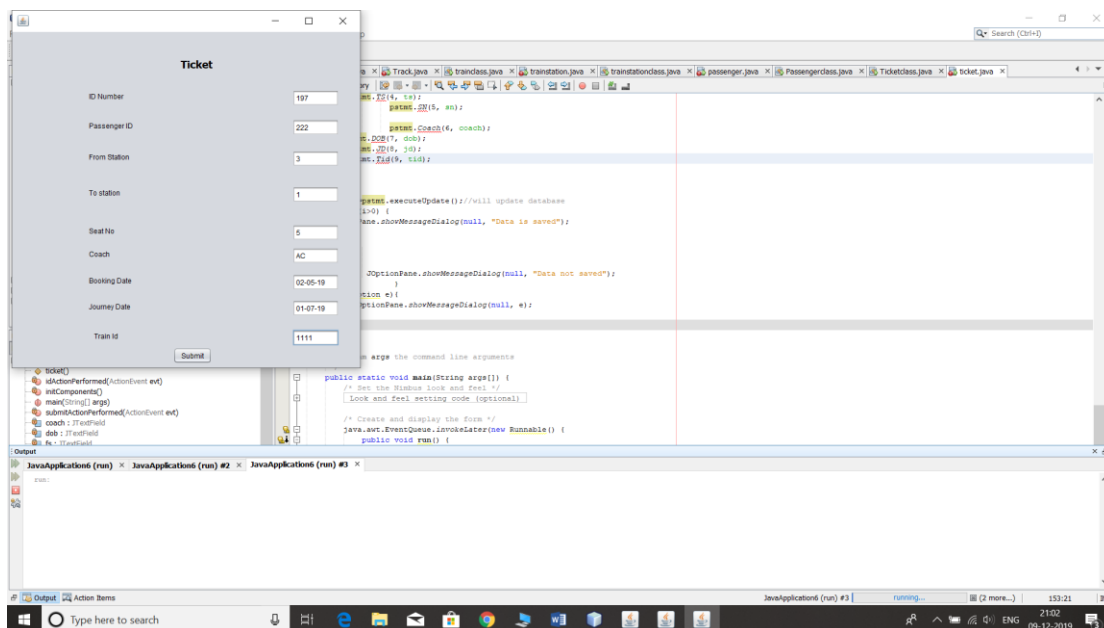
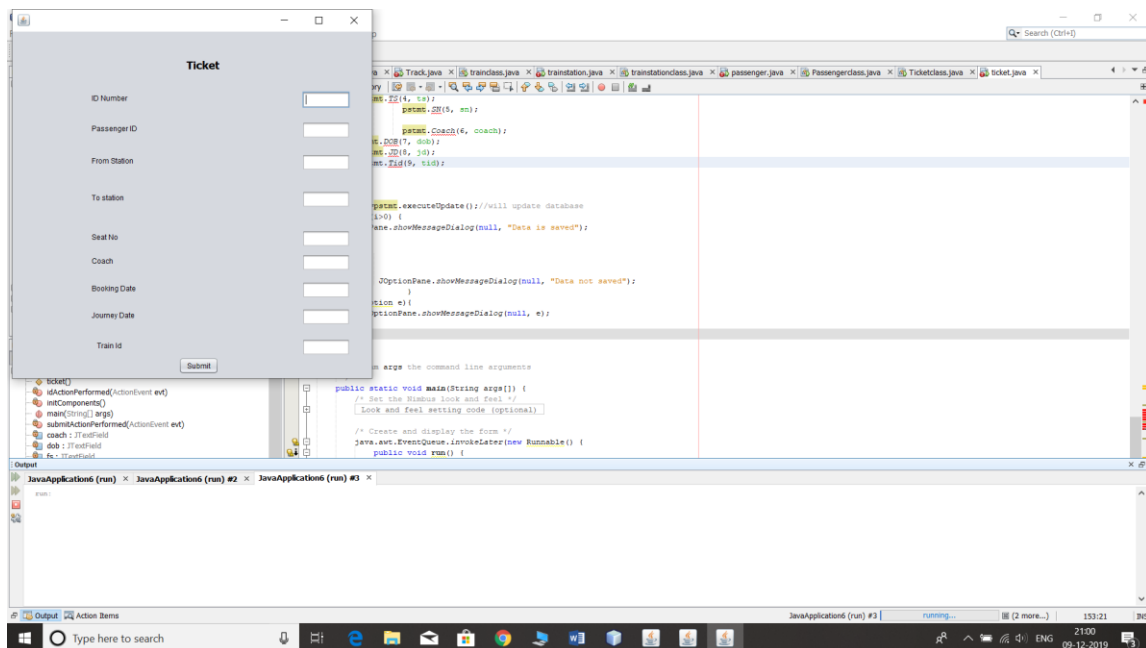
## Train Station:



Passenger:



**Ticket:**



## Conclusion:

As per the requirement the Database designing has been .I have used the mysql package available for connecting to the MySQL database.

I was able to create a MYSQL connection pool and query the data using the pool. It has been prepared which lets the users search for the trains and book tickets. For the front end part I have used java template and customized it to fit the needs. The form data was sent to the node service by making calls from front end using JQuery.

The stations names in the dropdown are also fetched directly from the database, which makes it easy to add new stations directly into the database.

Overall, I got good exposure on MySQL and NetBeans with this project.