Athlone Institute of Technology

Athlone

REPORT ON

CAR PARKING SYSTEM WITH HSQL DATABASE

Distributed System (Sem-2)

BE in Software Engineering

(4th year) Level-8 2019-2020

Submitted by:-

Deepansh Dubey

A00268737

Table of Contents

1 Introduction……3

2. Main Functions with Screenshot ……...4

3. Extra Functionality……5

4. Working of System……5

GUI Screenshots main page……6

GUI Screenshot Table Creation and fill table……6

Show Button ScreenShot……7

5. GUI After Insertion……..8

6. GUI After Updation ……9

7. GUI After Deletion……10

Delete By Id ……10

Delete All ……11

Stored Procedure Screenshot……..12

Excel sheet ScreenShot……...13

8. Database Maintainence……14

Warning Messages……..15

9. SQL Queries ……16

Creating the table ……..16

Inserting into the table……16

Updating the table……...17

Deleting the table……...17

Truncate the table……... 17

10. Conclusion……18

11. Lesson Learnt ……....18

12. Future Enhancement……..18

Report On Car Parking System

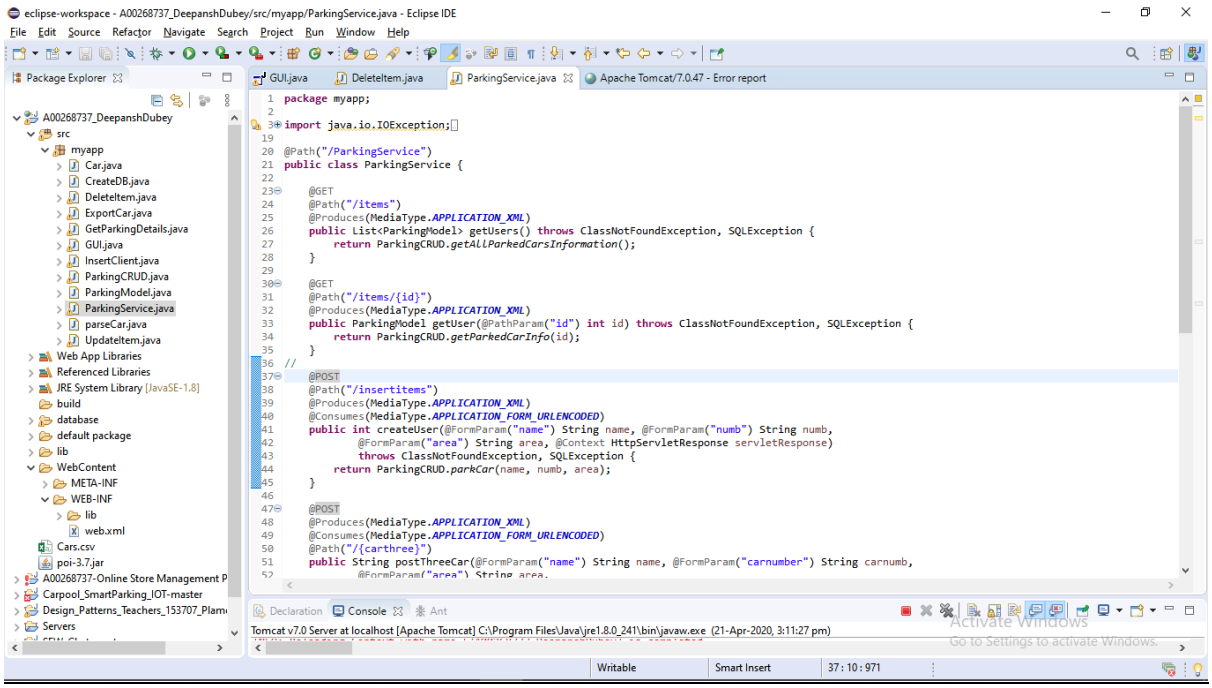
Title of the Project:- Car Parking System

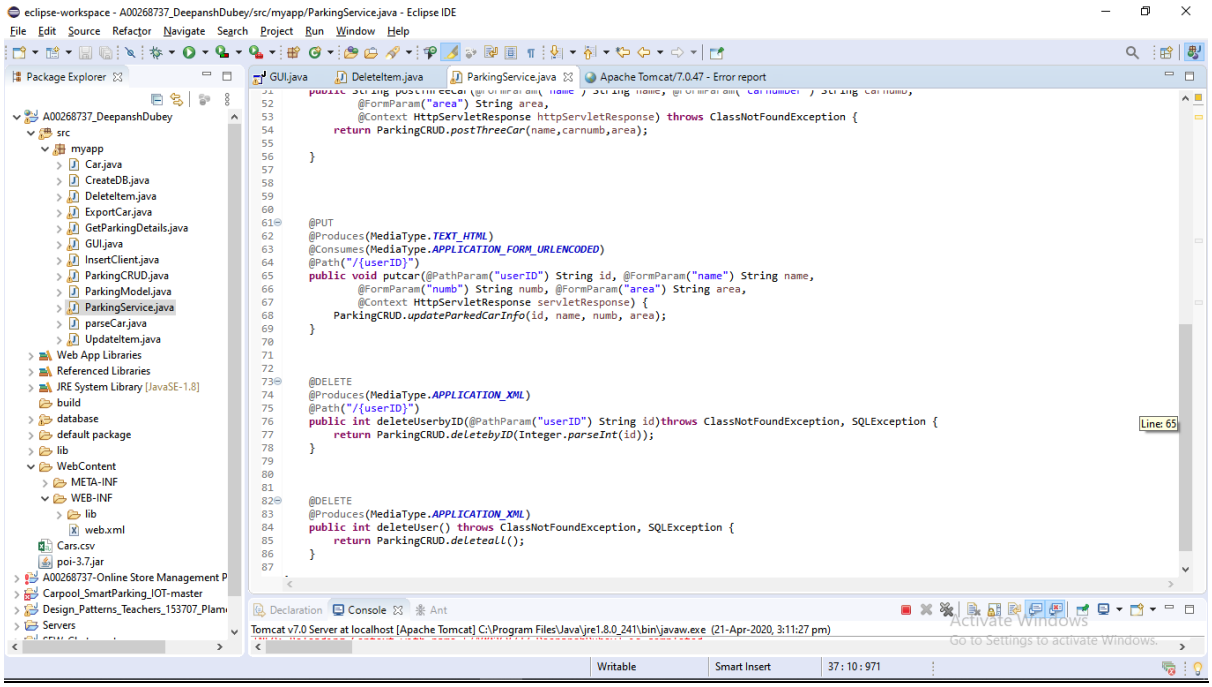
Introduction:-

This project implements the learning result in the Distributed System framework with what has been learned and taught. This project uses Rest service to incorporate client and server architecture. Make use of controller, layout, server and JAX api application class to view the details. A network program to communicate with the server where GET, POST, PUT and DELETE queries are created by the network. The project stores and reflects the information of Car parking running on Tomcat 7.0 server. HSQLDB stores the details and demand in the Cars table This project is focused on Car Parking to store the details regarding the Car parked in the parking area. This project makes management of the car parking facility more simpler and easier. It saves some important information when the car is parked as an ID, Owner Name of car, Car number, and the Area where it is parked. Information extracted is a method that uses XMLPullParser to identify tags.

Main Functions with Screenshots(Parking Service Class) :-

- **Get** – Created two Get function to get all data from the database table.
- **Put** – Enter the ID and can alter the other details and will be updated to the table.
- **Post** – Enter the details for a new Car and create it on the server and saves to database.
- **Delete All -** Removes all data from the tables and reset the identity by using truncate table query.


eclipse-workspace - A00268737_DeepanshDubey/src/myapp/ParkingService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
GUI.java
DeleteItem.java
ParkingService.java
Apache Tomcat/7.0.47 - Error report
A00268737_DeepanshDubey
src
myapp
Car.java
CreateDB.java
DeleteItem.java
ExportCar.java
GetParkingDetails.java
GUI.java
InsertClient.java
ParkingCRUD.java
ParkingModel.java
ParkingService.java
parseCar.java
UpdateItem.java
Web App Libraries
Referenced Libraries
JRE System Library [JavaSE-1.8]
build
database
default package
lib
WebContent
META-INF
WEB-INF
lib
web.xml
Cars.csv
poi-3.7.jar
A00268737-Online Store Management P
Carpool_SmartParking_IOT-master
Design_Patterns_Teachers_153707_Plam
Servers
package myapp;
import java.io.IOException;
@Path("/ParkingService")
public class ParkingService {
@GET
@Path("/items")
@Produces(MediaType.APPLICATION_XML)
public List<ParkingModel> getUsers() throws ClassNotFoundException, SQLException {
return ParkingCRUD.getAllParkedCarsInformation();
}
@GET
@Path("/items/{id}")
@Produces(MediaType.APPLICATION_XML)
public ParkingModel getUser(@PathParam("id") int id) throws ClassNotFoundException, SQLException {
return ParkingCRUD.getParkedCarInfo(id);
}
//
@POST
@Path("/insertitems")
@Produces(MediaType.APPLICATION_XML)
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
public int createUser(@FormParam("name") String name, @FormParam("numb") String numb,
@FormParam("area") String area, @Context HttpServletResponse servletResponse)
throws ClassNotFoundException, SQLException {
return ParkingCRUD.parkCar(name, numb, area);
}
@POST
@Produces(MediaType.APPLICATION_XML)
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Path("/{carthree}")
public String postThreeCar(@FormParam("name") String name, @FormParam("carnumber") String carnumb,
Declaration
Console
Ant
Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (21-Apr-2020, 3:11:27 pm)
Activate Windows
Go to Settings to activate Windows.
Writable
Smart Insert
37 : 10 : 971


eclipse-workspace - A00268737_DeepanshDubey/src/myapp/ParkingService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
GUI.java
DeleteItem.java
ParkingService.java
Apache Tomcat/7.0.47 - Error report
@FormParam("area") String area,
@Context HttpServletResponse httpServletResponse) throws ClassNotFoundException {
return ParkingCRUD.postThreeCar(name,carnumb,area);
}
@PUT
@Produces(MediaType.TEXT_HTML)
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Path("/{userID}")
public void putcar(@PathParam("userID") String id, @FormParam("name") String name,
@FormParam("numb") String numb, @FormParam("area") String area,
@Context HttpServletResponse servletResponse) {
ParkingCRUD.updateParkedCarInfo(id, name, numb, area);
}
@DELETE
@Produces(MediaType.APPLICATION_XML)
@Path("/{userID}")
public int deleteUserbyID(@PathParam("userID") String id)throws ClassNotFoundException, SQLException {
return ParkingCRUD.deletebyID(Integer.parseInt(id));
}
Line: 65
@DELETE
@Produces(MediaType.APPLICATION_XML)
public int deleteUser() throws ClassNotFoundException, SQLException {
return ParkingCRUD.deleteall();
}
Declaration
Console
Ant
Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (21-Apr-2020, 3:11:27 pm)
Activate Windows
Go to Settings to activate Windows.
Writable
Smart Insert
37 : 10 : 971

Extra Functionality:-

- **Stored Procedure :-** Through out the method the parameter is passed or transferred to insert the data into the respective tables accordingly. It helps in storing the data into the tables.

- **Jtable Scrollbar :-** I attached a scrollbar to my Jtable as well, but it's not available due to limited entries.

- **Warning Messages :-** I have also added warning messages in this application if the textfield are empty .

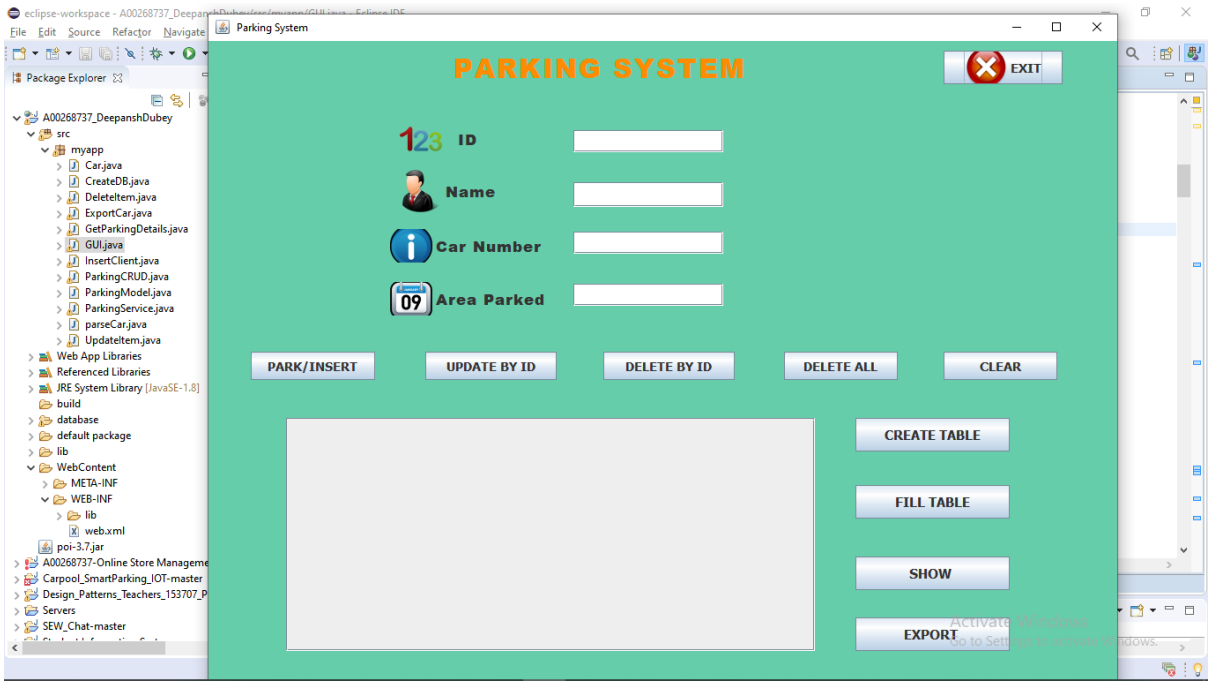
- **Print to Excel sheet :-** Prints all the details from table to an Excel sheet (CSV format – Comma separated values).

- **Truncate :-** To delete all data by the help of Truncate Query and reset by its identity.

Working Of System:-

1. **This is main Page called GUI.java .**

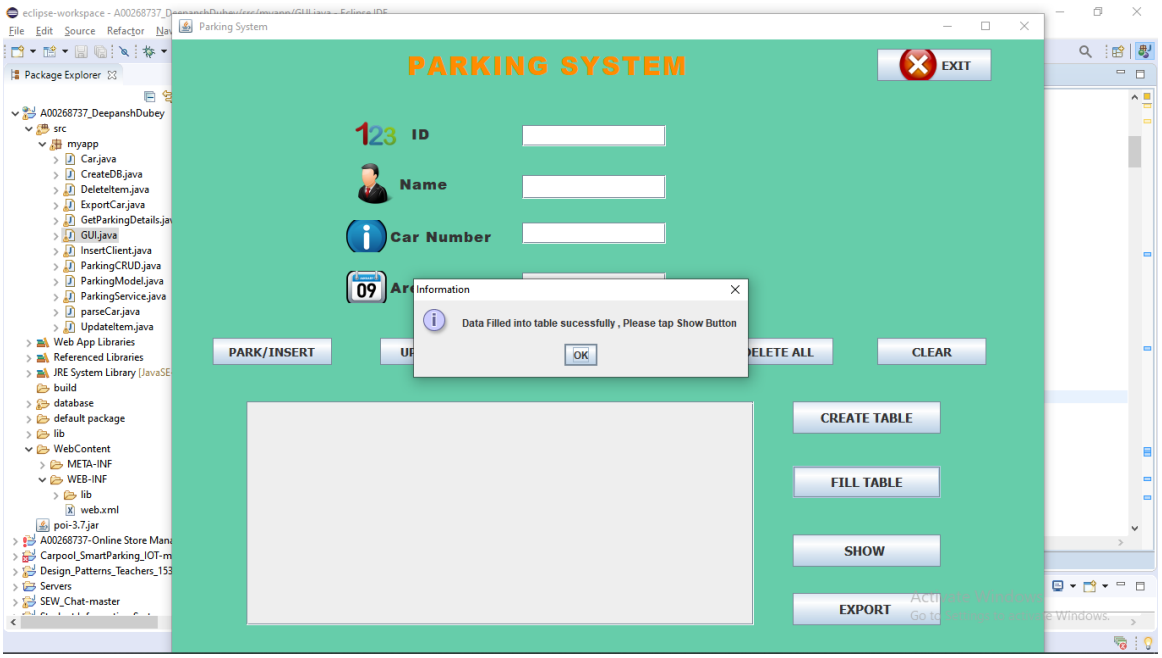
   GUI Screenshot Main Page



2. **Create Table after clicking Button called ‘Create table’.**

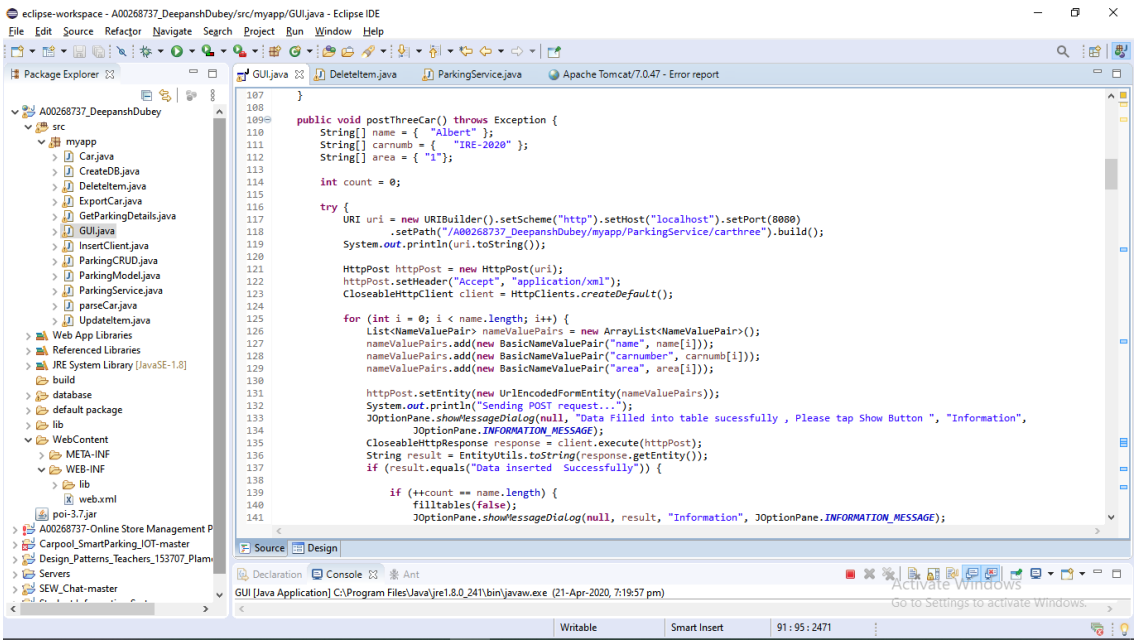
GUI Screenshot Table Creation

3. **FILL TABLE :- After Clicking the FillTable Button , it will insert the dummy data.**



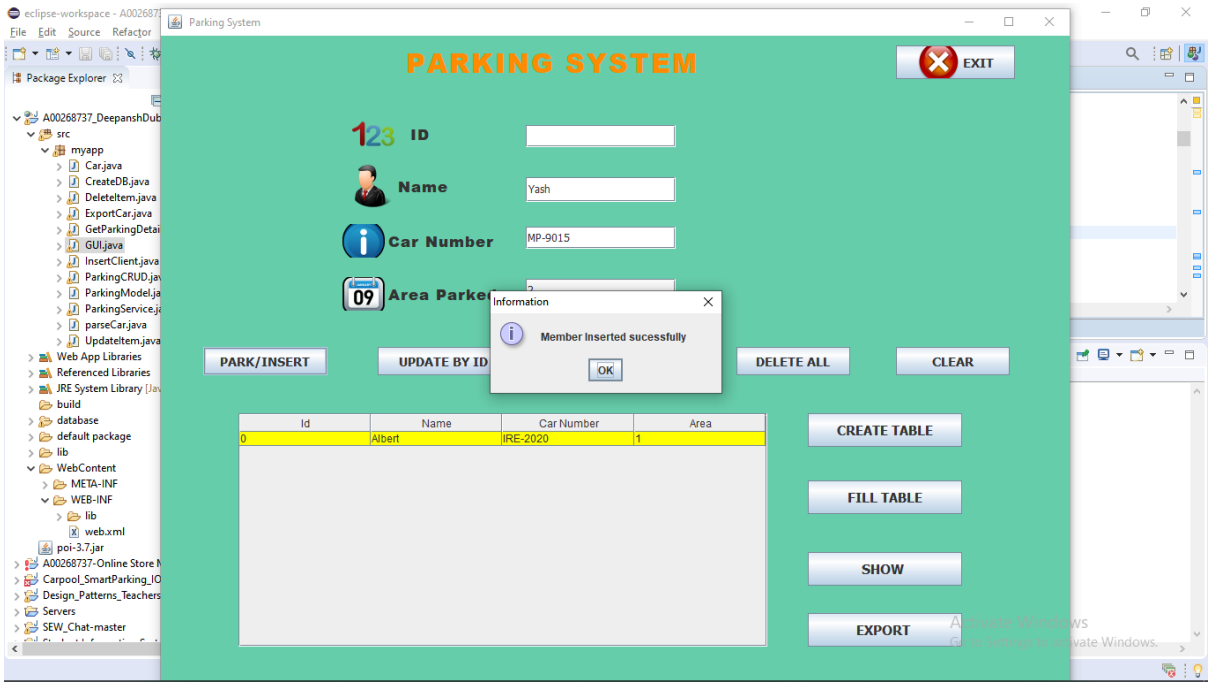
4. **SHOW :- This button will show the Dummy data into the Jtable .**

I had added dummy data in the Code.

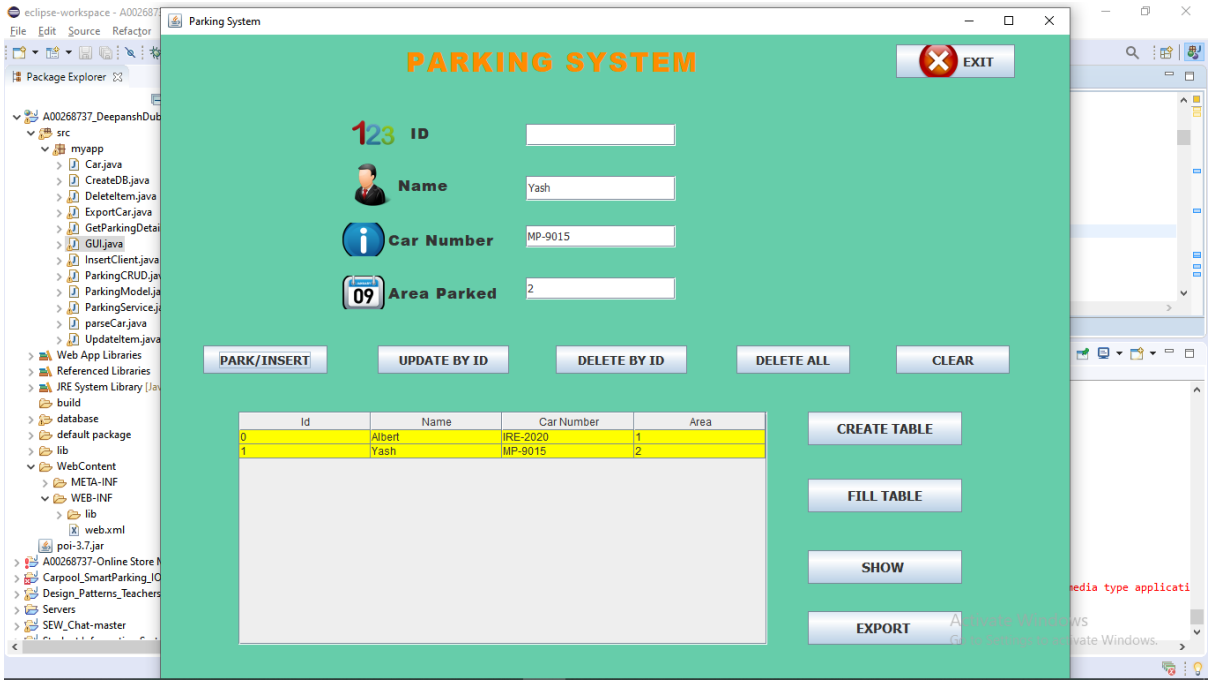


GUI AFTER INSERTION:-

After filling the textfield and inserting all the data display the data clicking on Park button and display in JTable.

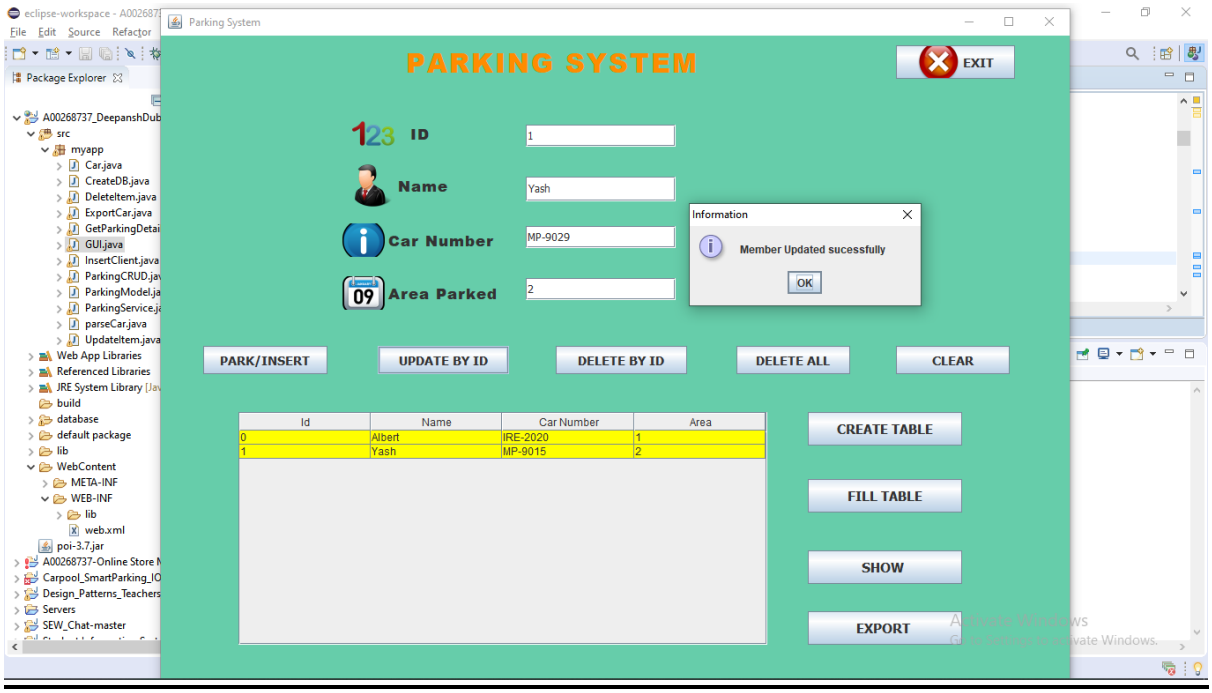


Inserted Into the table

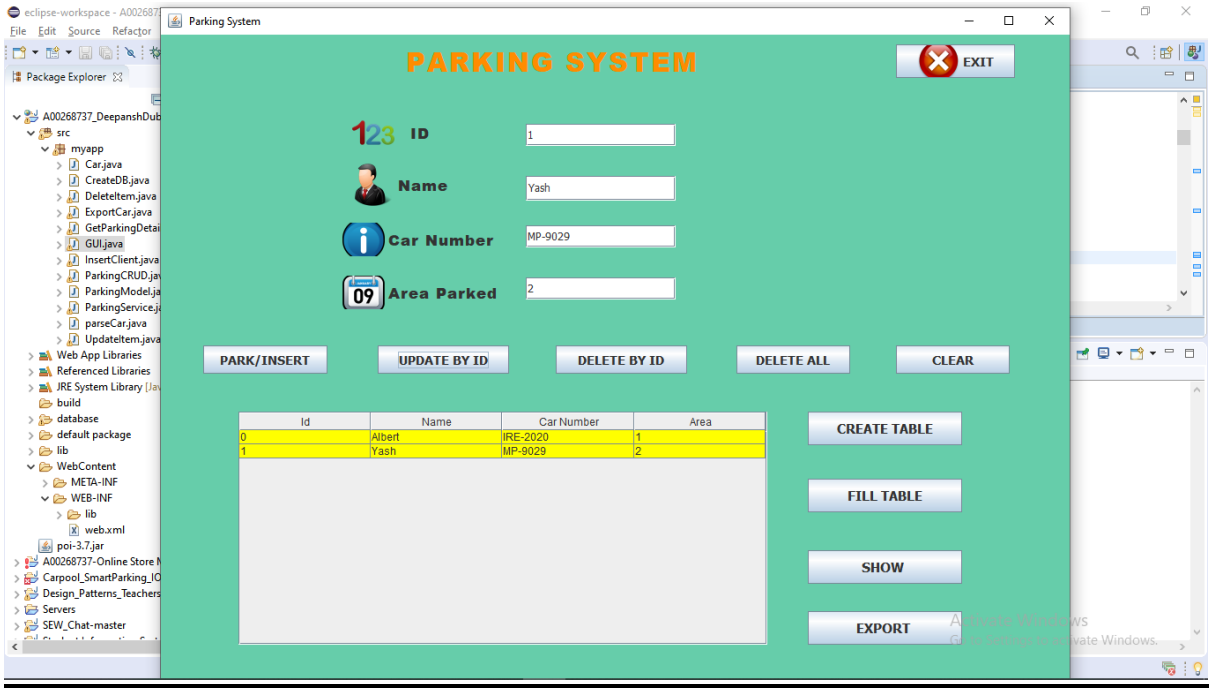


GUI AFTER UPDATION:-

This Update (Button) method accepts the parameter needed to update, in this the important parameter is ID, which is used to update the data. The ID acts as the reference for the update method and allows the changes to other respective fields.



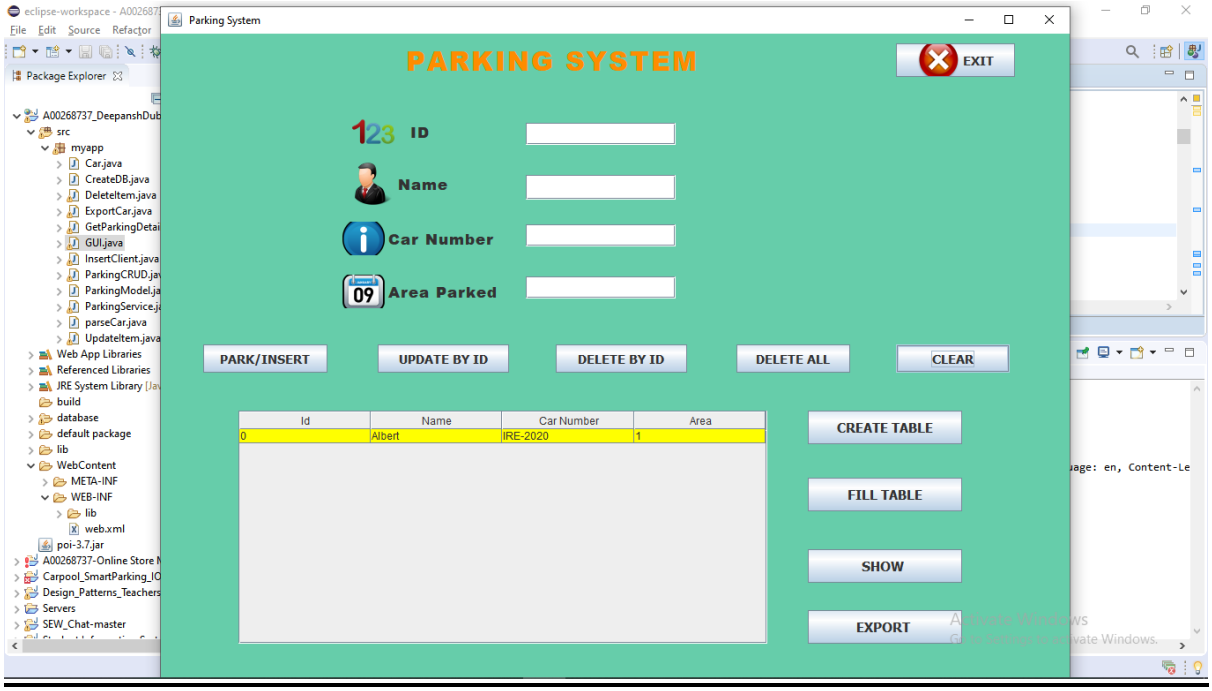
Car Number is Updated



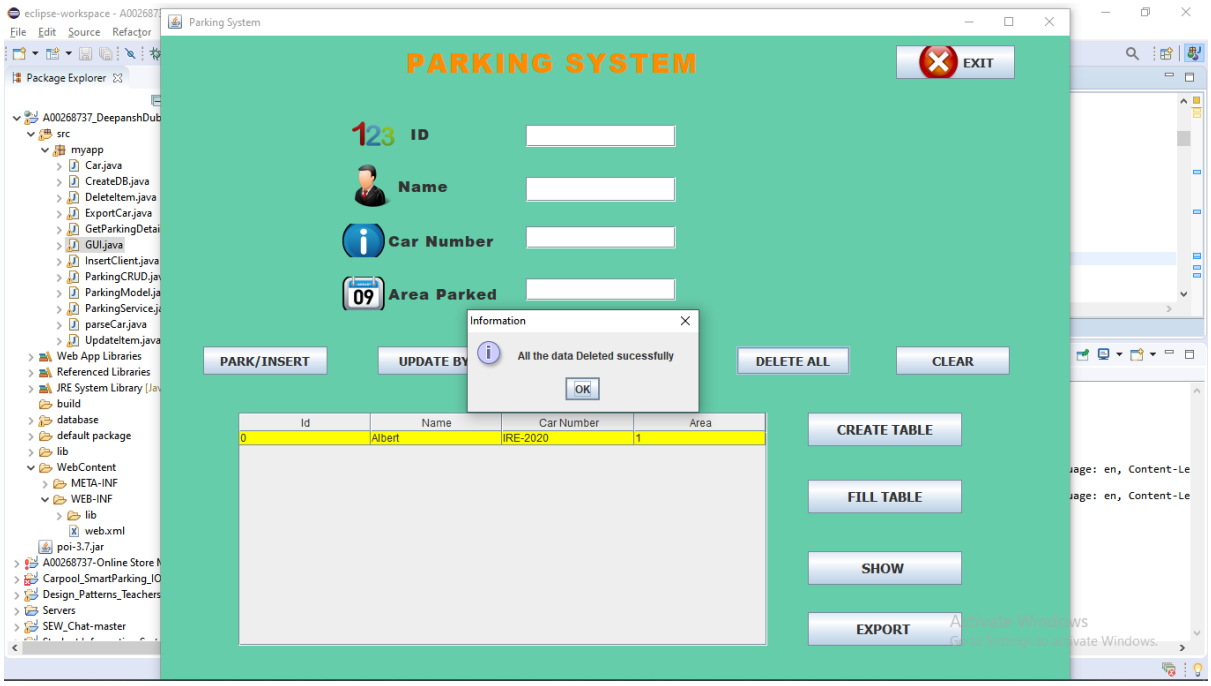
GUI AFTER DELETION:-

Delete By ID :- Delete by Id would delete the data by entering the Car ID and eventually update it to the Jtable.

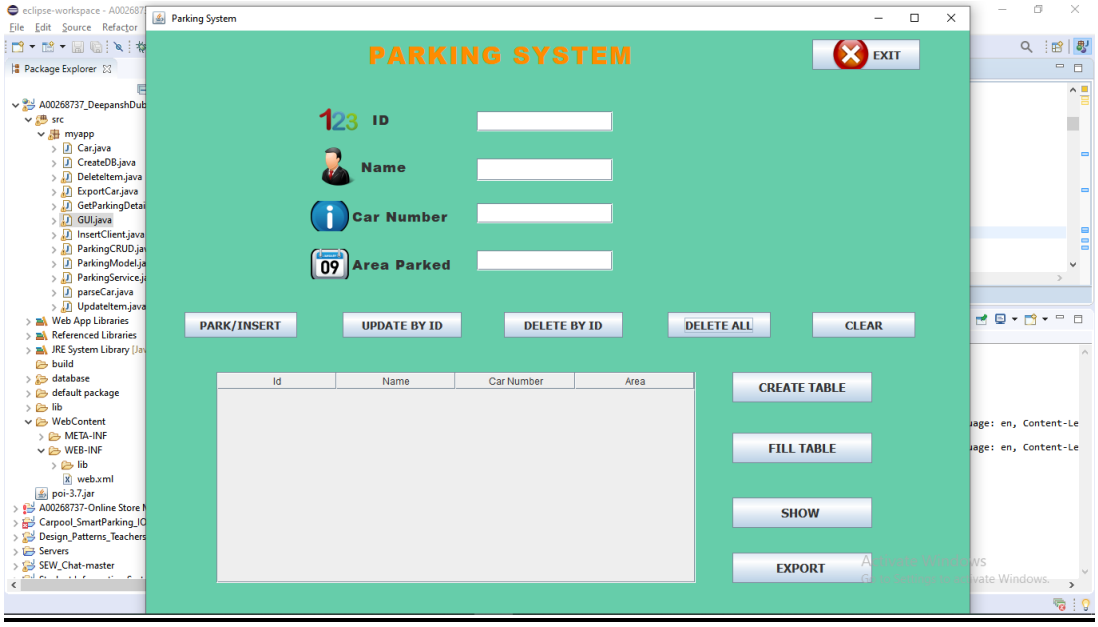
After Deletion by particular ID ;-



Delete all :- Delete all the data by clicking Delete All button with the help of Truncate Query and reset by its identity. And After inserting the data id will starts from 0.



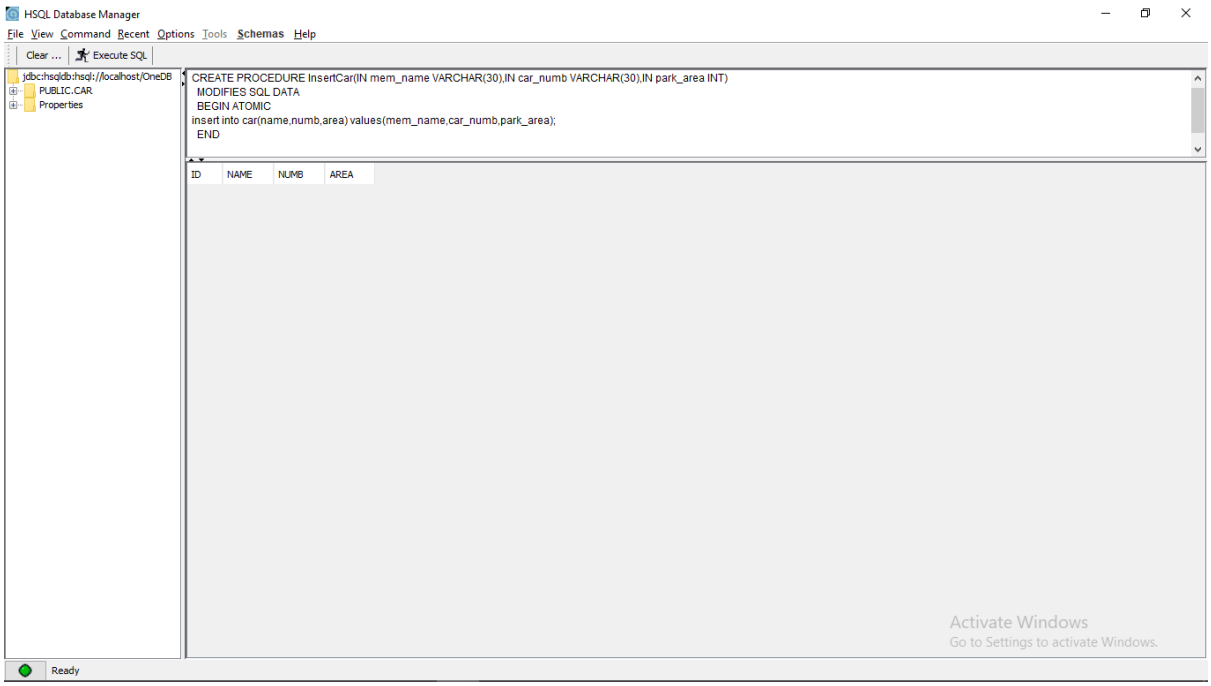
After Deletion :- Deleted all the data from the table



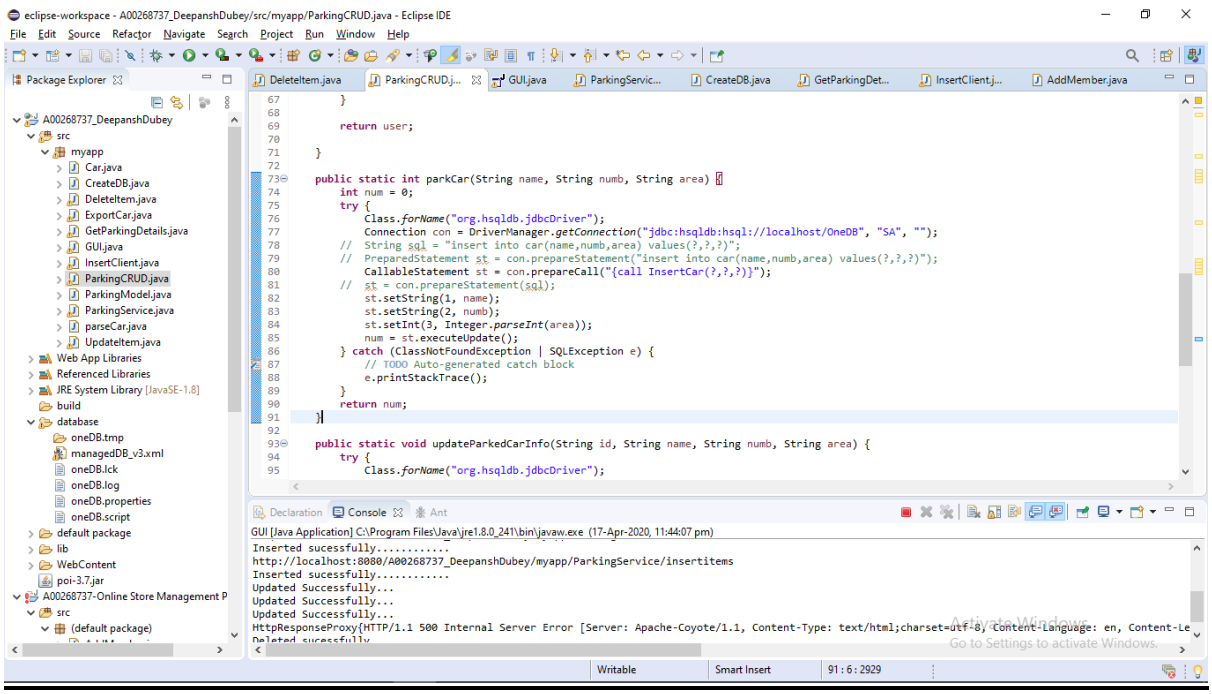
Stored Procedure :-

Using Stored Procedure to insert the data in the table . Passing parameters in Insert button function and it automatically inserts data in the table and fetch the data in Jtable.

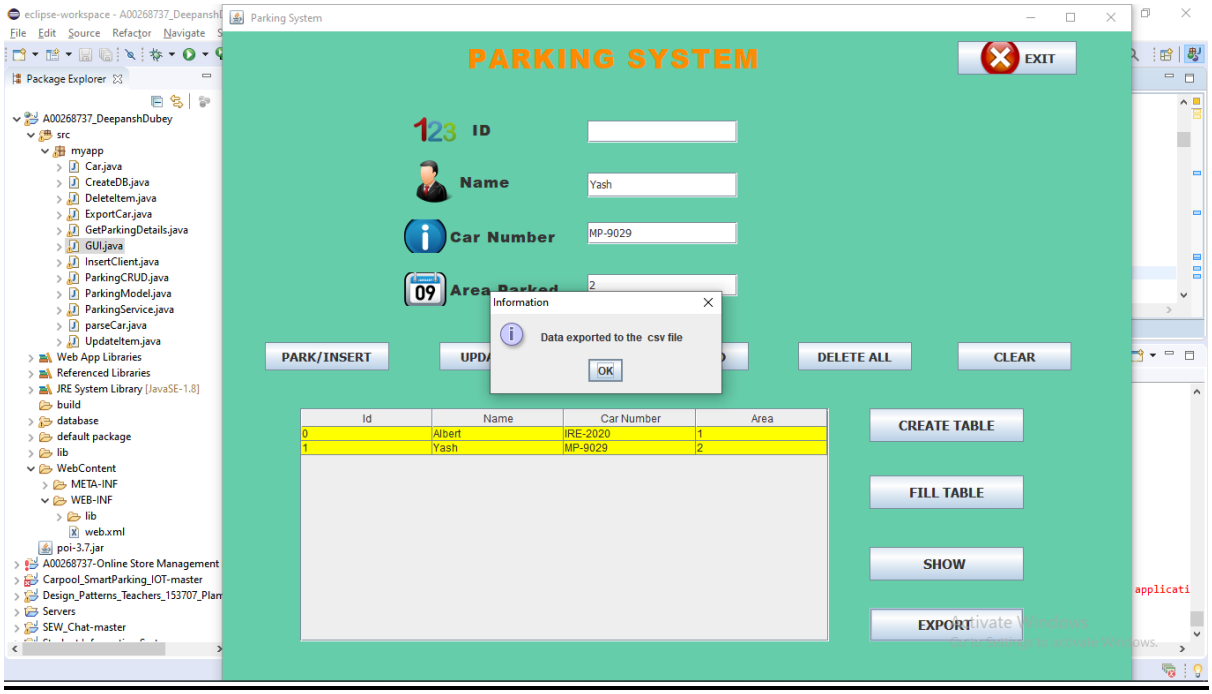
STORED PROCEDURE QUERY



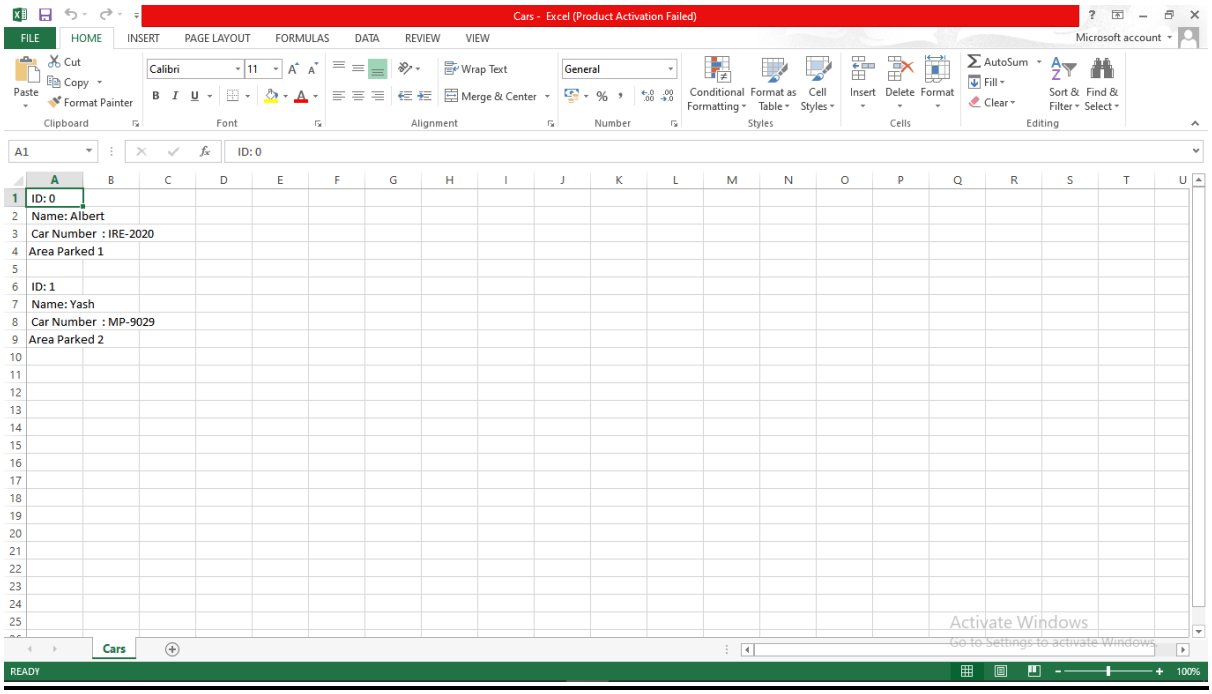
Calling the Stored Procedure on Insert button query (Park button)



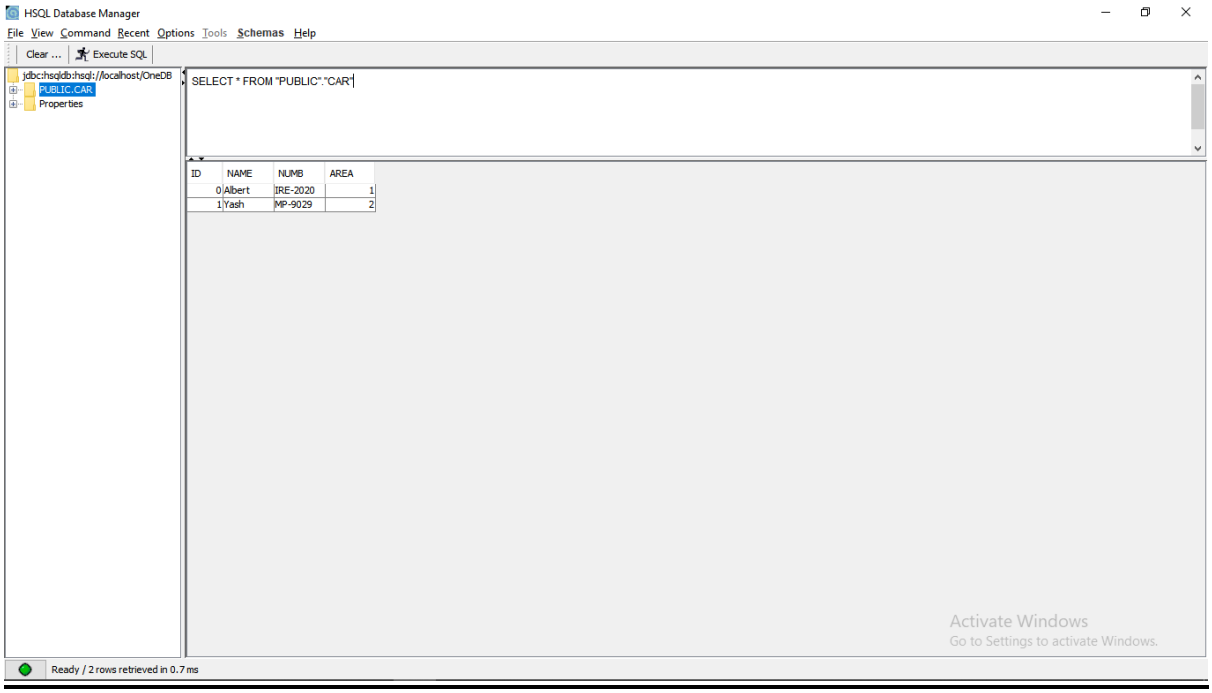
Export Data into Excel sheet(CSV) :-



Excel Sheet Screenshot:-



Database Maintainence:-

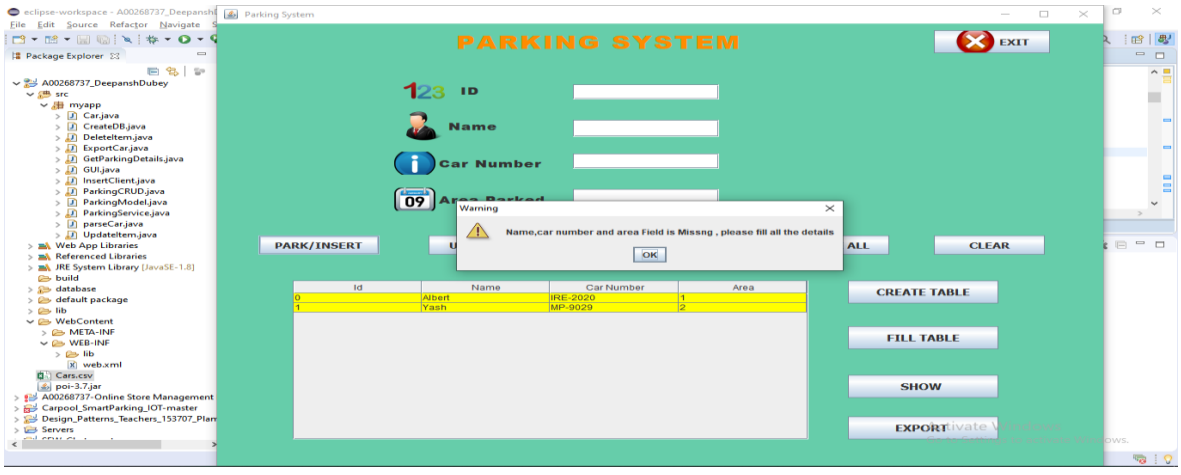


Maintain the database:-

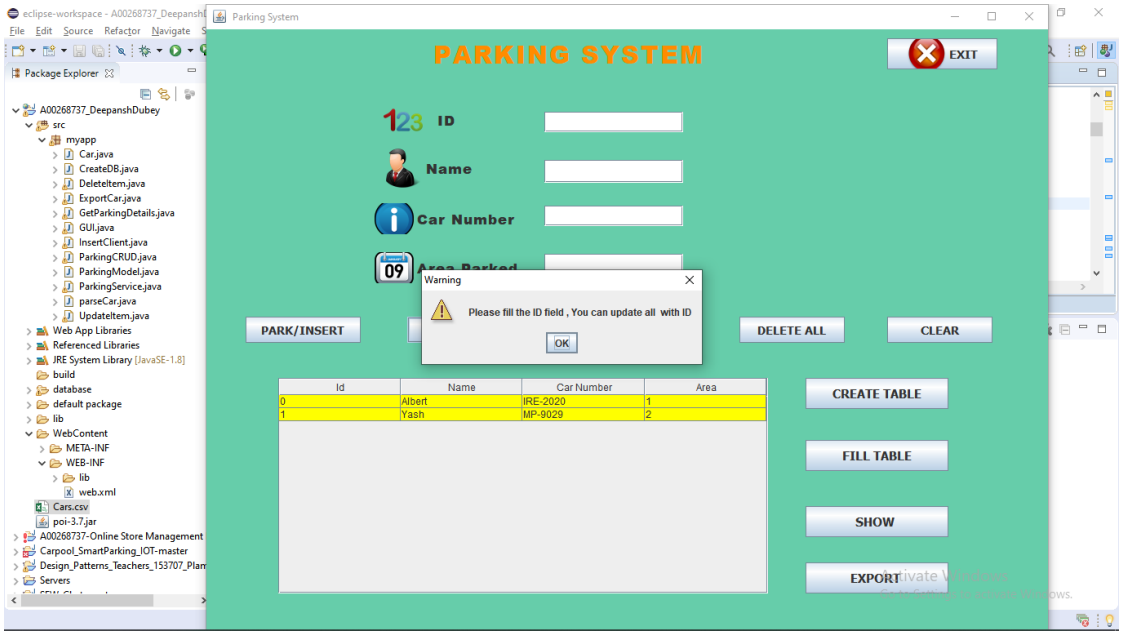
1. Run the Database Ant File provided on moodle
2. Run the Server and the table will be active.
3. On ant file press manage to handle the database manually
4. Run GUI.java in Client to interact with the database

Warning Messages ScreenShots :-

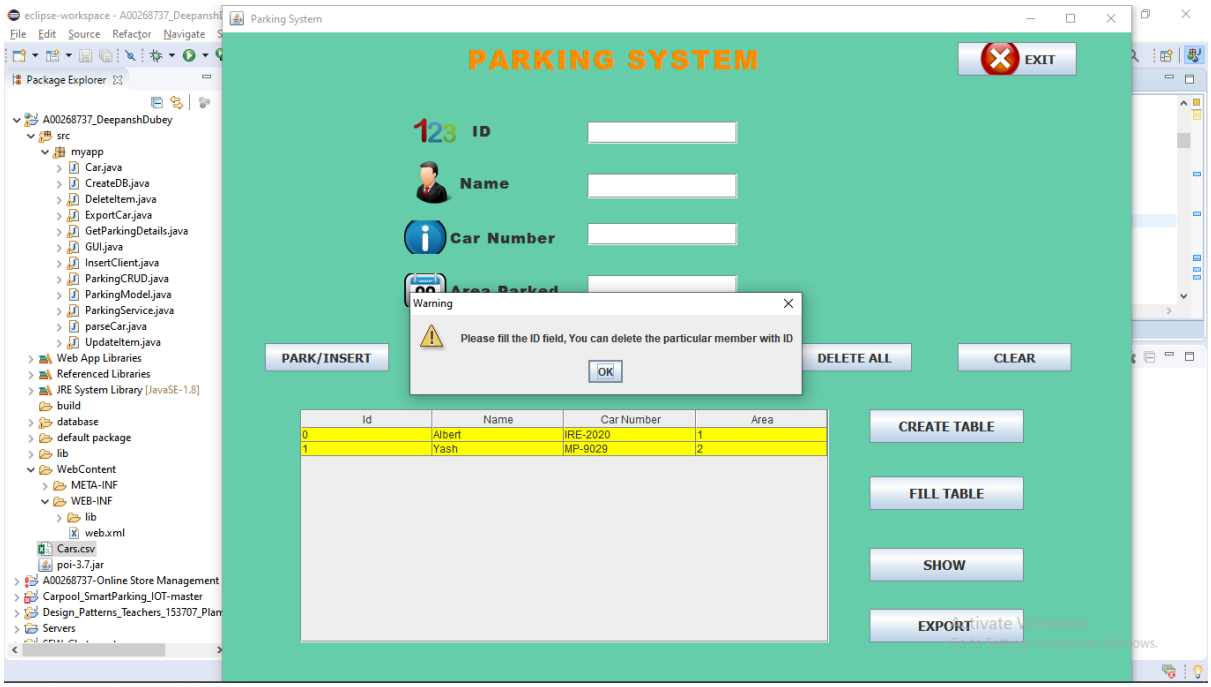
- When you just click the Park/insert Button without filling all the text field it will show the error or warning message



- When you just click the Update Button without filling the text field of ID , it will show the error or warning message .

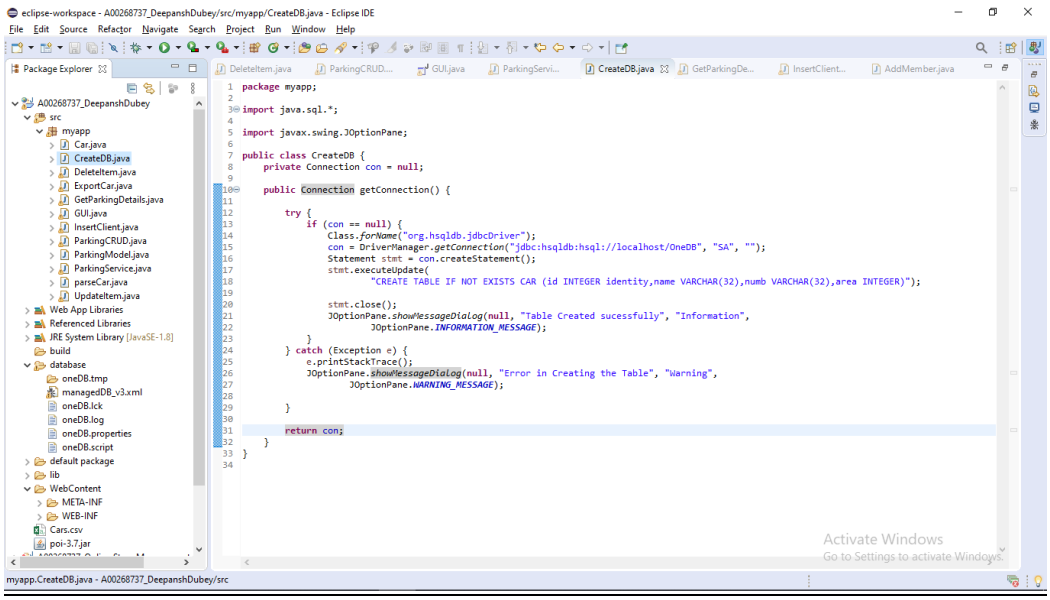


- When you just click the Delete Button without filling the text field of ID , it will show the error or warning message .

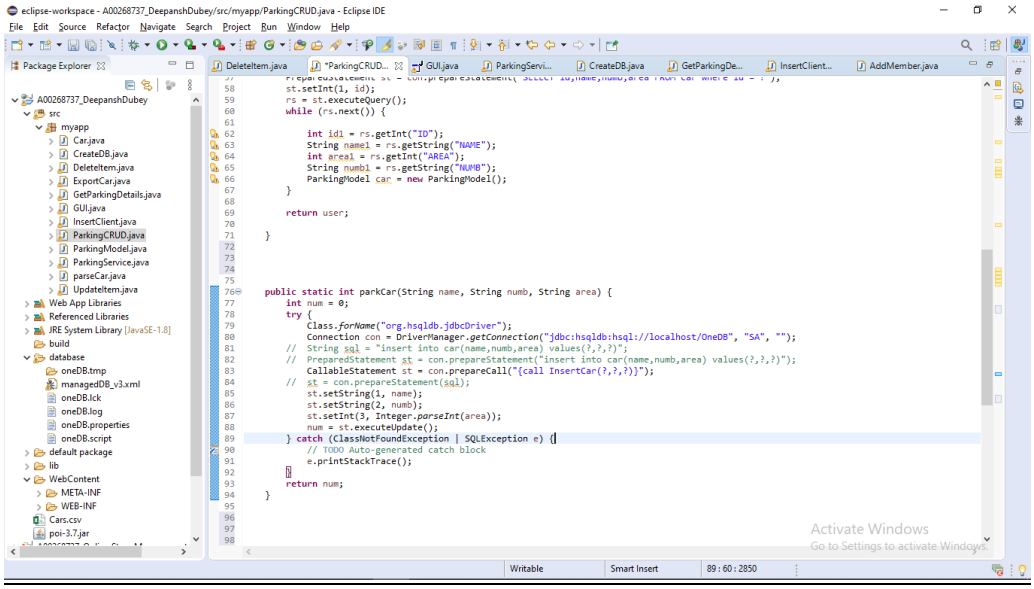


SQL QUERIES:-

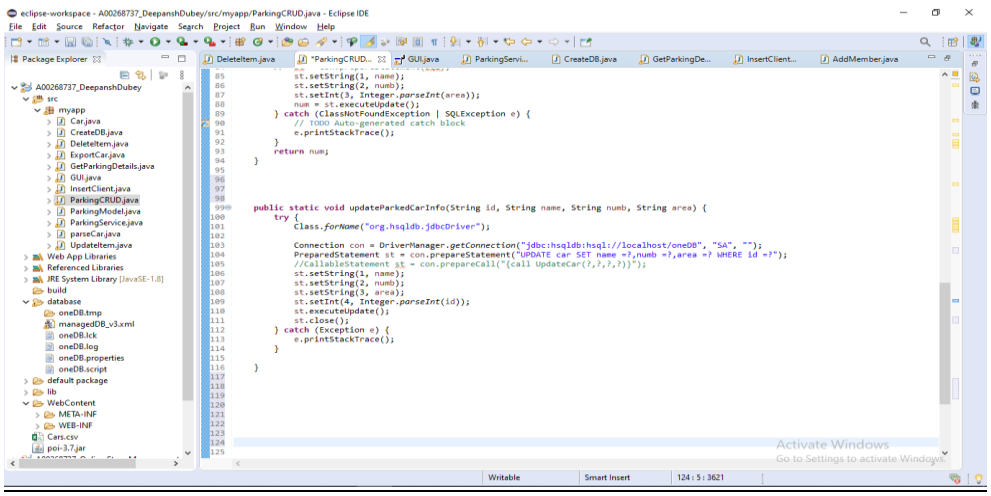
❖ Creating the table:-



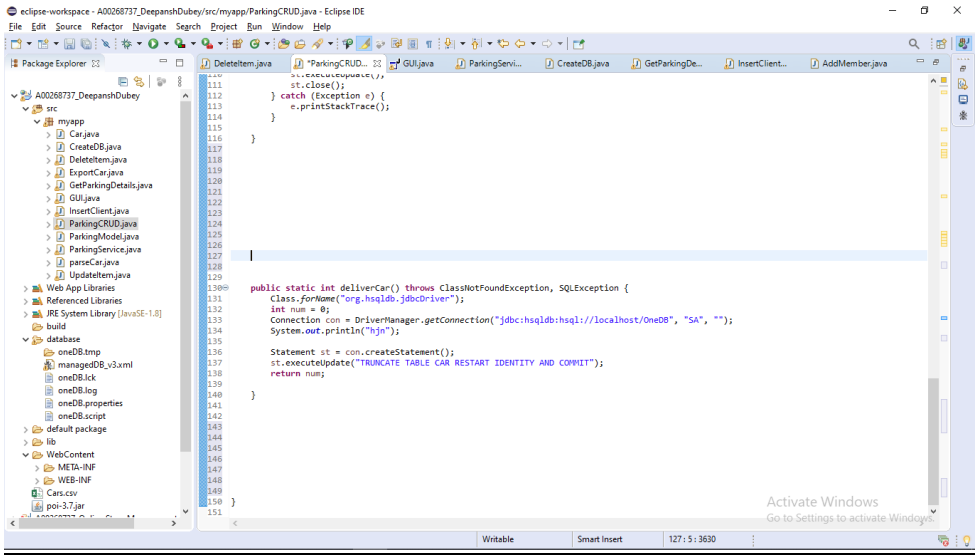
❖ Inserting into the table:-



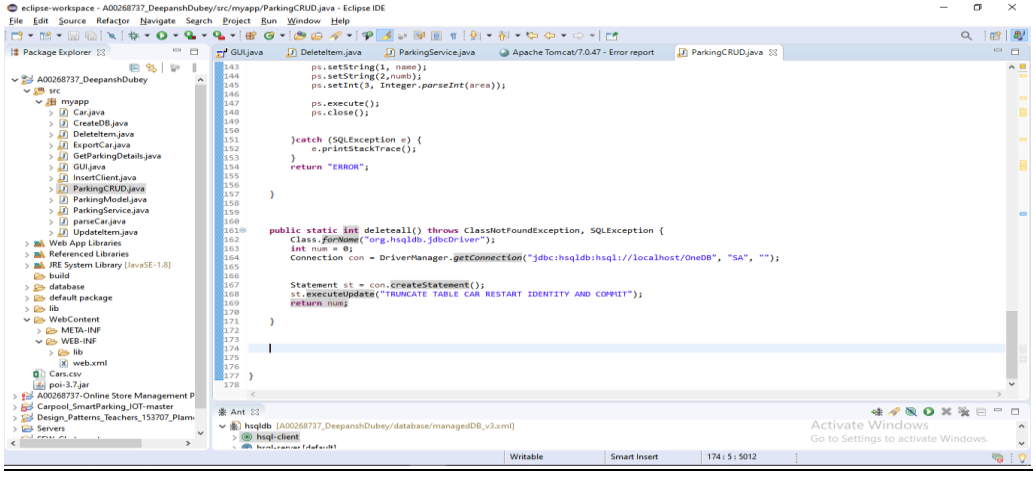
❖ Updating the table:-



❖ Deletion from the table:-



❖ Truncate the table (Delete All):-



Conclusion:-

This project has fulfilled all the requisite specifications for both client and server-side applications. The client-side version of the project uses XMLPullParser to view the results in the GUI. And all the data that these two programs handle is running on the HSQLDB. I am successful in making the application where we can Add, delete, show save the details as per the requirement. Through it has some limitations , I strongly believes that the implementation of this system will surely benefit the organisation. I had a wonderful time designing a prototype that was a perfect reconstruction of all the laboratories we worked on. In the immediate future, I would love to collaborate on related ventures.

Lesson Learnt:-

Doing something for long time periods always gives good lesson. Some of the things that I had learnt are as follows:-

- Learnt about the REST API Framework.
- Learnt about the Java Swing technology, its component and ways to implement them.
- Learnt to work in pressure and to be patient.
- Learnt to manage the data in file format(Excel sheet).
- Assignment helped me to improve the knowledge of REST application and connection between client and server.
- Overall this assignment helped me a lot to improve my understanding of implementing REST application.

Future Enhancement:-

Since this assignment was started with very little knowledge about the Car Parking System, we came to know about the enhancement capability during the process of building it. Some of the scope we can increase for the betterment and effectiveness are as follows:-

- Interactive User Interface Design
- Use of Oracle as its database.
- Online Payment can be added.
- Make the system flexible in any type.

Google drive Link Screencast

https://drive.google.com/open?id=1UVdkxss2-Sr_b7h7oODkaiXHsefosWlk