# Abstract

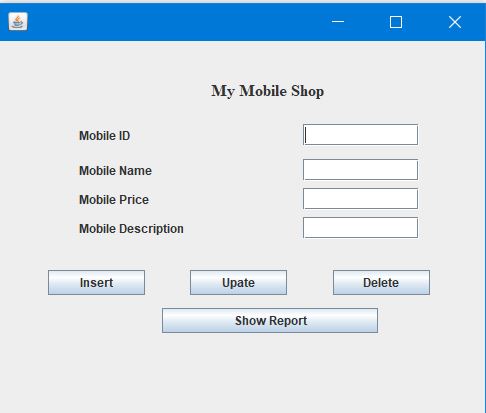
This project implements the learning outcome of what all has been gained and learnt in distributed system module. This project implements client and server architecture with the use of rest service. Making using of controller, model and a service class in server and a client application to interact with the server where client makes GET,POST, PUT and DELETE requests.

# GUI:

Client app is a single page application made with swing

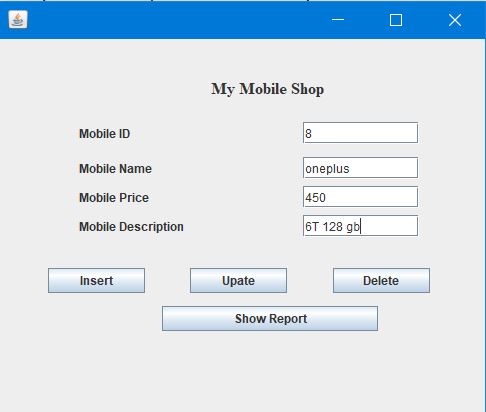
## Screenshots:

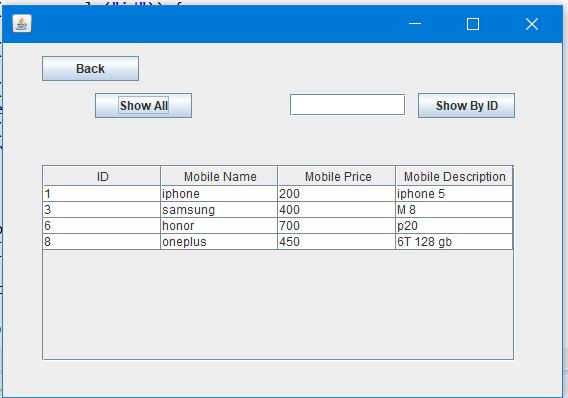
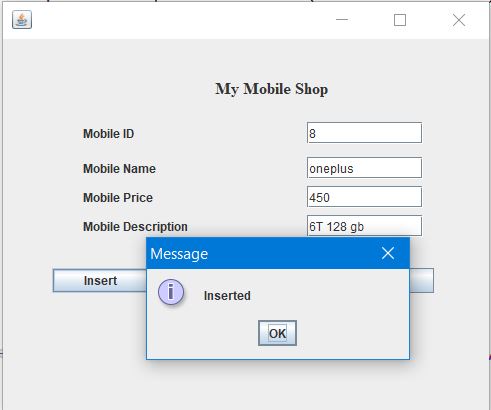
### Main Page:

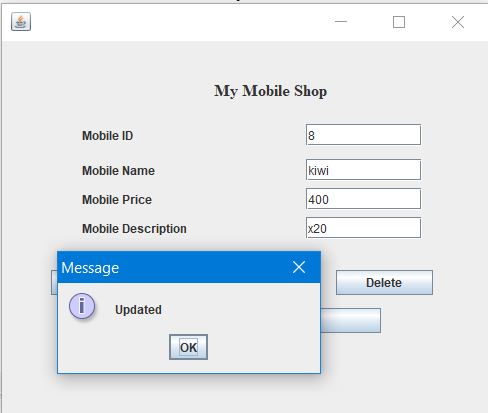
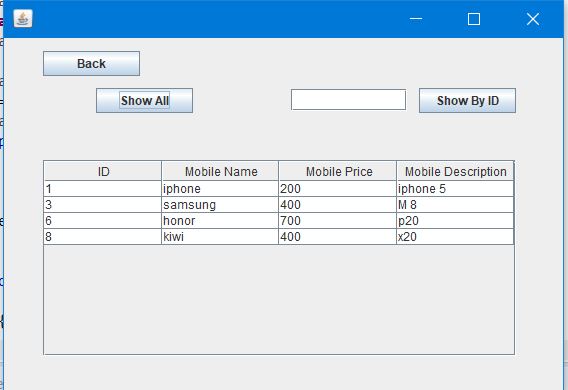


This is the main page which opens up when the application is started, this page has all the functionalities and is interacting with the server directly.

#### Insert Mobile Details Page:

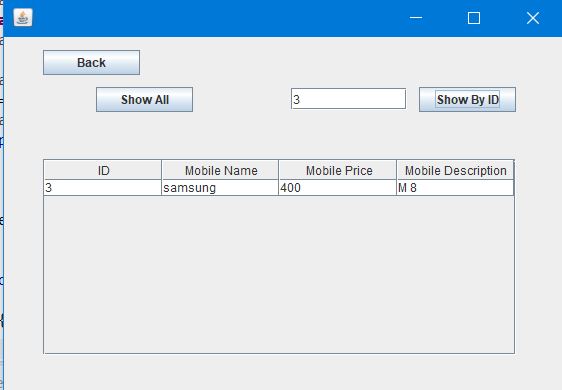
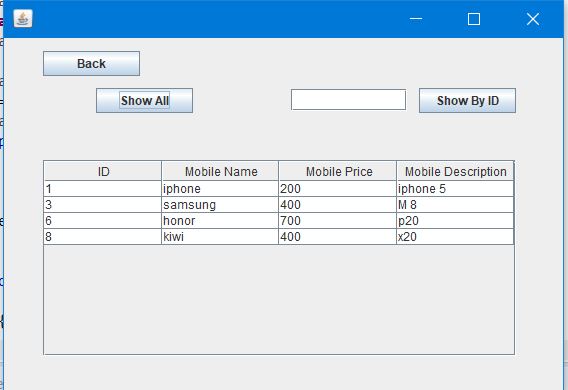
Here is how the data is inserted into the system by using post method



Update Mobile Details (PUT): ****

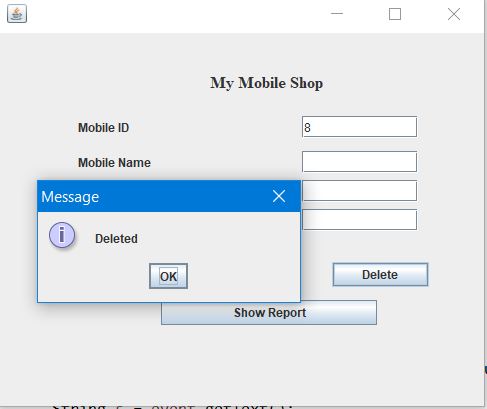
Based upon the data that already exists in the system we can take the id and update the associated details with update button using PUT method.

#### Show Details (GET):

****

In this page, the user has two options where he can chose to display the details by Id or display the all the details that are available in the mobile table with the use of GET.

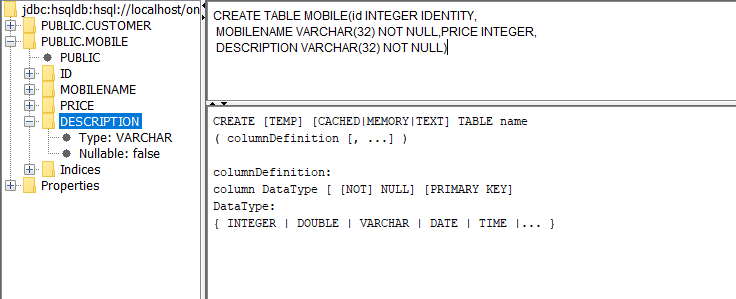
#### Delete Mobile Details (DELETE):

****

In this page the user can delete a particular mobile from the table by using the ID.

## Table Creation:

Query used to create the table:



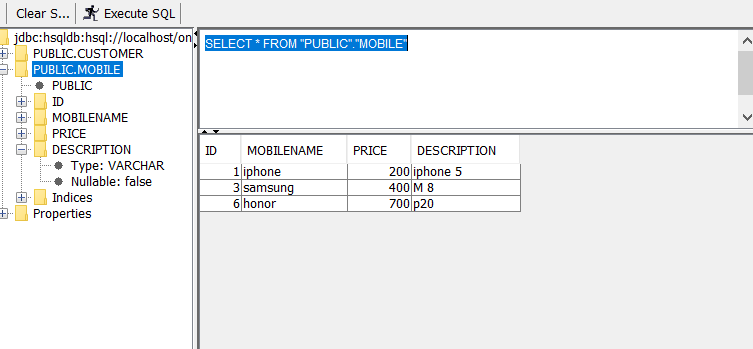
CREATE TABLE MOBILE(id INTEGER IDENTITY,

MOBILENAME VARCHAR(32) NOT NULL,PRICE INTEGER,

DESCRIPTION VARCHAR(32) NOT NULL)

**Data in the Mobile Table:**

Select \* from MOBILE;

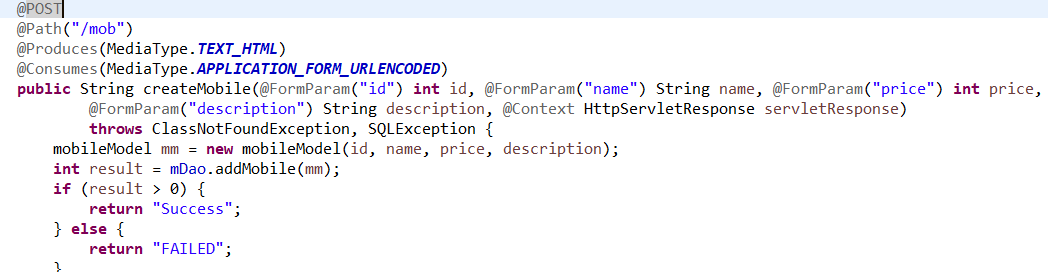


### Maintain the database

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 main.java in Client to interact with the database**

## Method from server Side

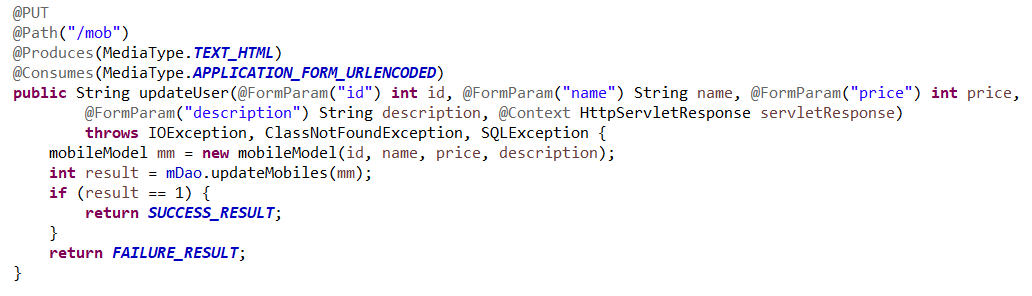
### POST:



This method accepts the parameter needed to insert in the database and returns expressesion of the success or failure of the insertion process. This method is called at the GUI with the respective parameters.

PUT**:**

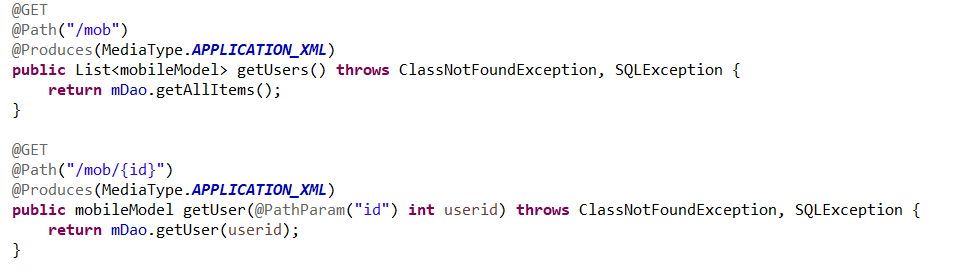
In the client application, the PUT function is written in the update the mobiles



This 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.

### GET:

In the client application, the PUT function is written in the DefaultTable class, which holds two methods named,



The getUser method returns all the data in the table as a TableModel which is used to populate the JTable in the GUI.