A picture containing drawing

Description automatically generated

**SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY**

DISTRIBUTED SYSTEM

(SE3020)

YEAR 3 SEMESTER 1 | GROUP 8

FIRE ALARM SENSOR MONITORING SYSTEM

REPORT

|  |  |
| --- | --- |
| **STUDENT ID** | **STUDENT NAME** |
| IT18181548 | SUTHARSHAN.S |
| IT18186970  IT18188882  IT18164886 | ASHVINI.P  UDDIN.R  RUPASINGHE.D.M.H.M |
|  |  |

# Content

1. Introduction………………………………………………………………………………3
2. High level architectural diagram……………………………………………….4
3. Workflow diagram…………………………………………………………………….5
4. System workflow scenario execution………………………………………..6
5. Appendix – source codes and binaries

# Introduction

This fire alarm monitoring system is implemented using REST API, web client, dummy sensor application, RMI server and client. the technologies which are used to implement the system as follows;

* REST API - spring boot
* web client - react, bootstrap,axios
* sensor application - react
* RMI server and client - java
* PhpMyAdmin is used for store sensor and user details.

The system administrator registers the sensors, and when the level of co2 and smoke rises, the relevant sensors will be activated.

The fire alarm monitoring system has a web client and a desktop client that can display sensor data such as room number, floor number, co2 level, smoke level and active / inactive status for both users and admin.

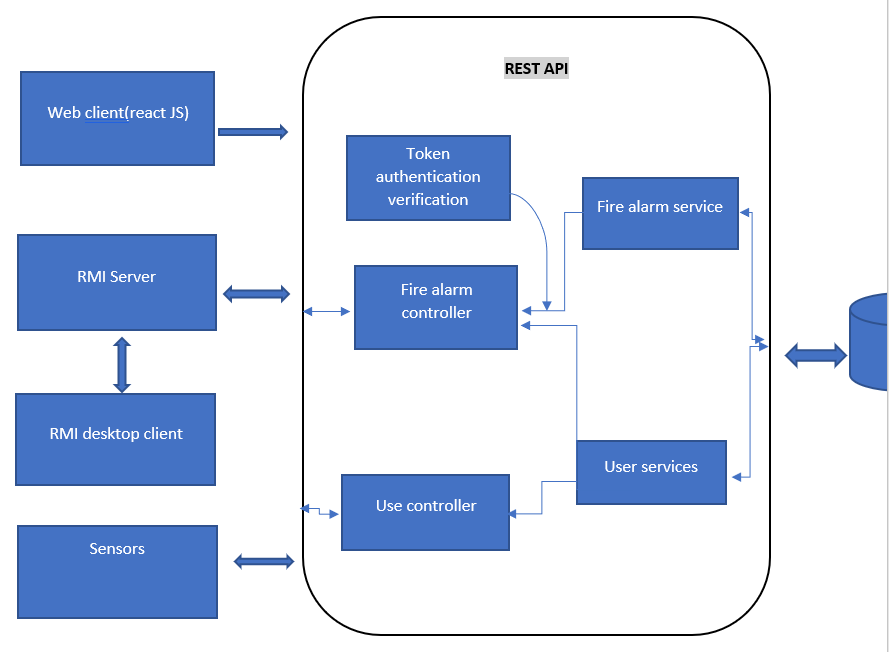
A dummy alarm sensor application is implemented to manually control the level of co2 and smoke. The Admin will receive an email from the system when the sensor is activated.

The structure and the process of each component of the project is explained below;

A picture containing object, clock

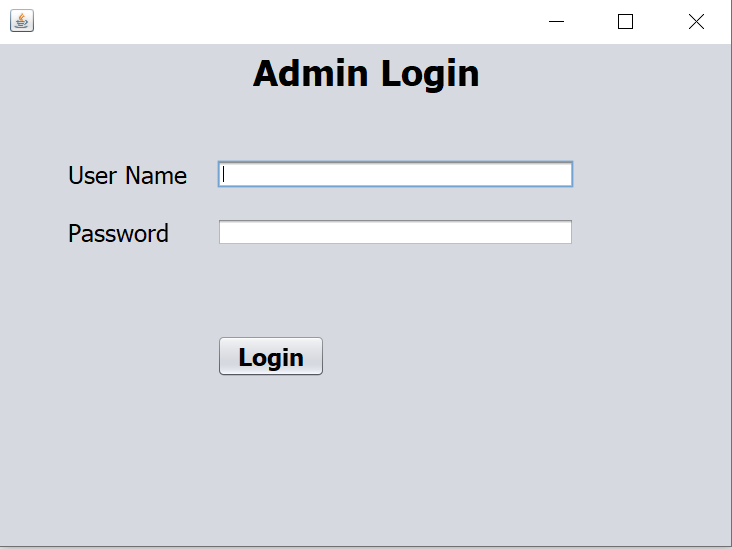
Description automatically generated

# **High level architectural diagram for fire alarm monitoring system**

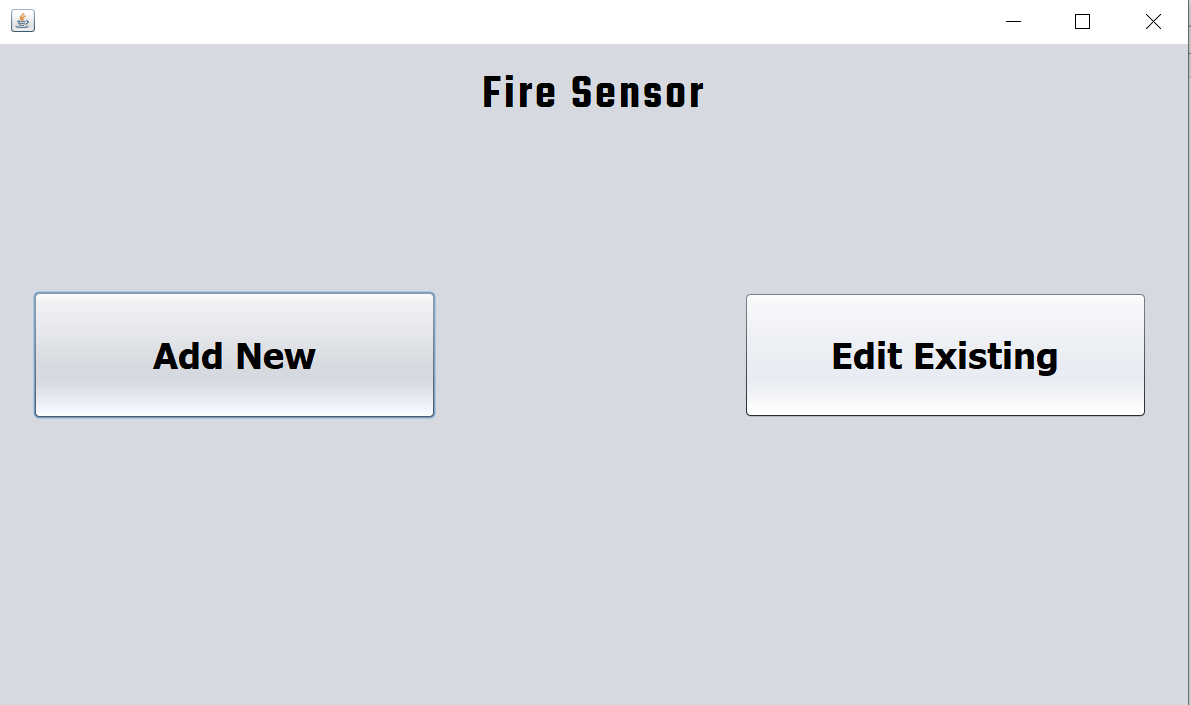


# **A screenshot of a cell phone Description automatically generatedWorkflow diagram**

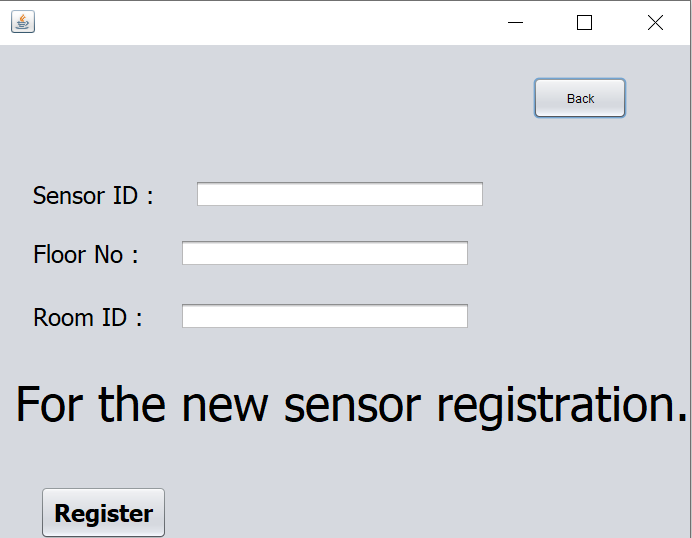
# **System workflow scenario execution**



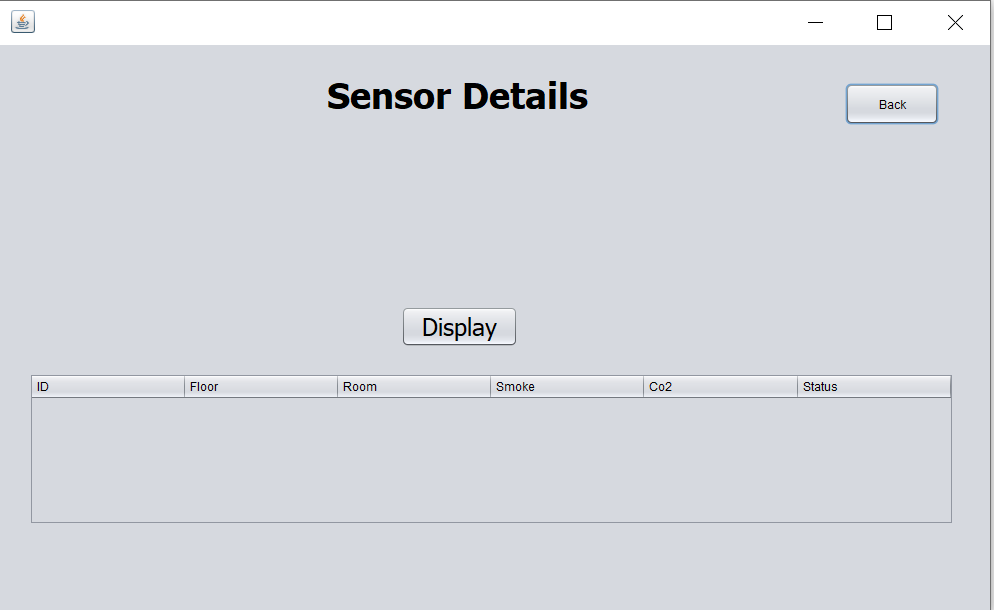
Admin should log in to the system using login credentials in the above figure and he can add a new sensor for the building entering the relevant details by clicking the ADD NEW button as in the following figure.



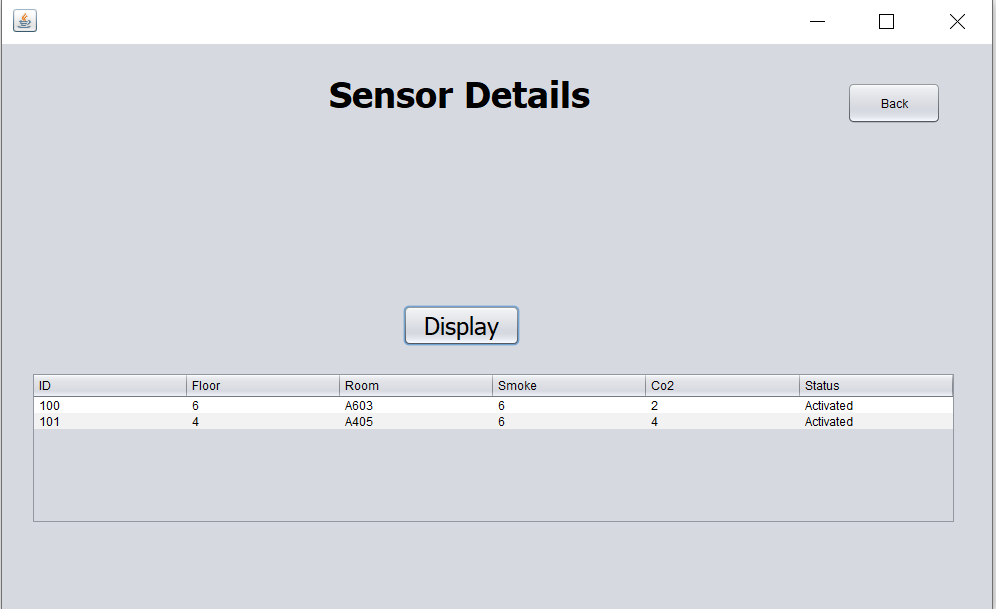
Admin can add new sensor details through this diagram,



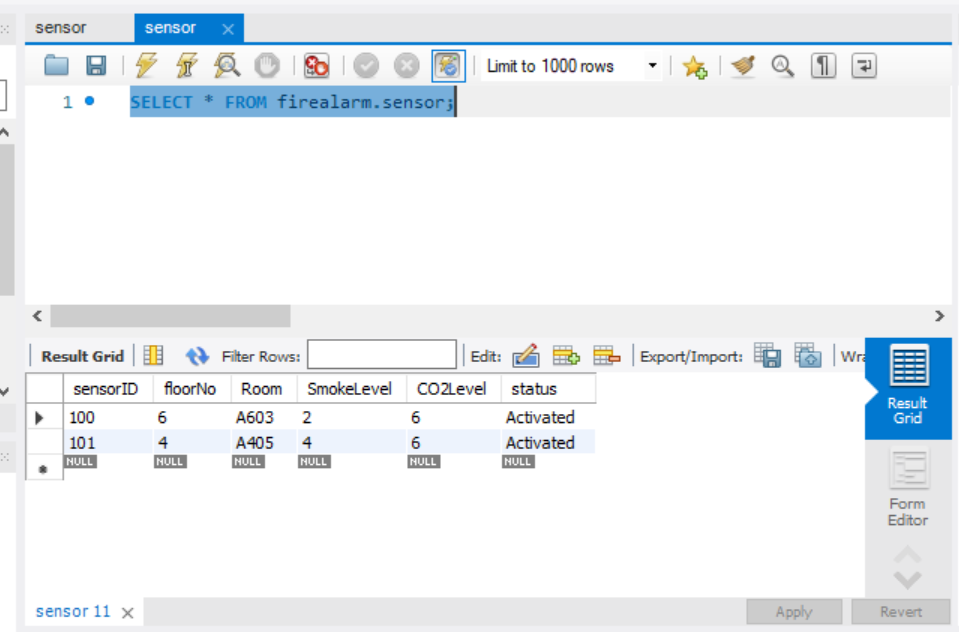
Also Admin can Edit the sensor details from the database through the REST API .



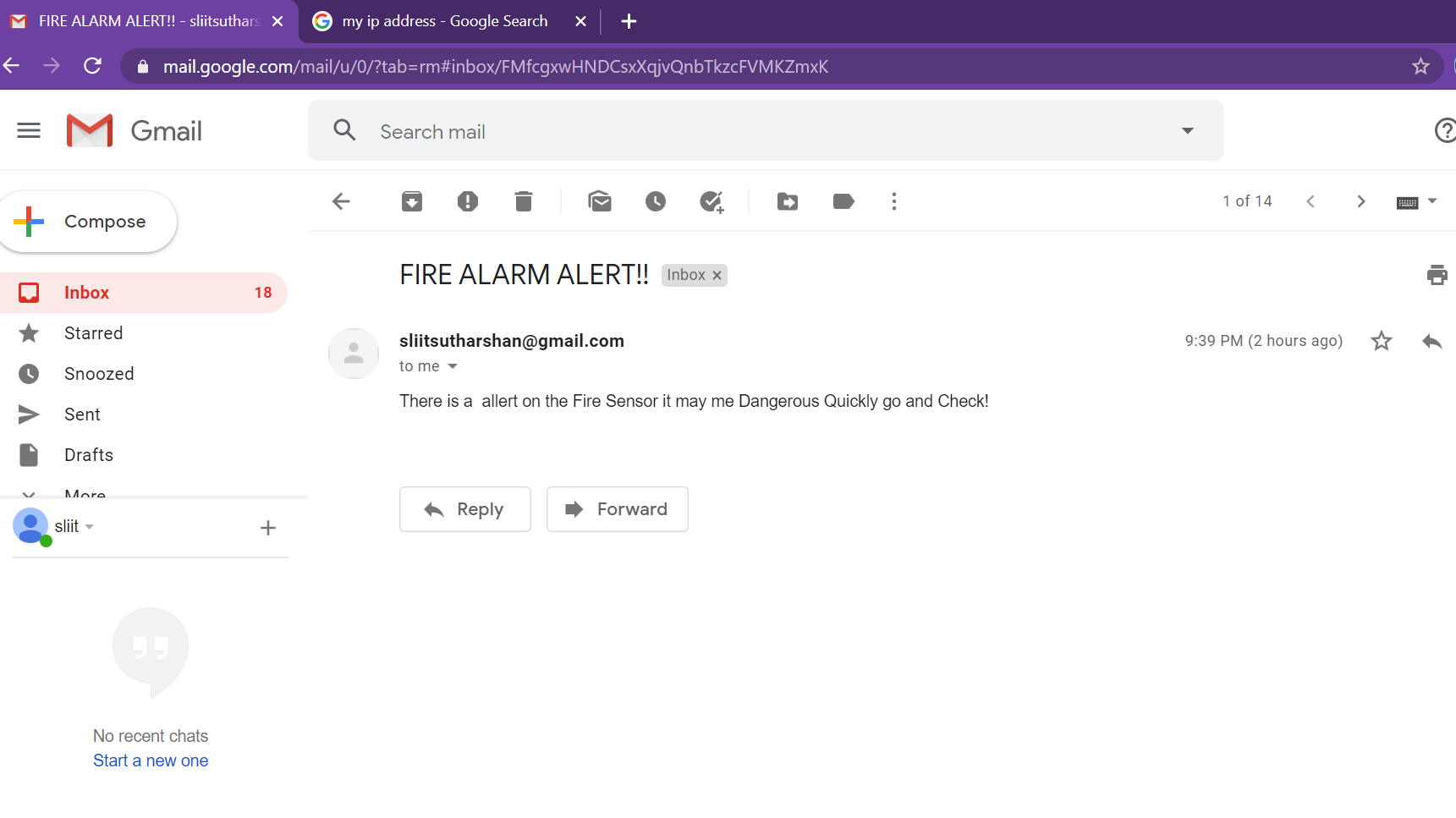
It should display as follows,



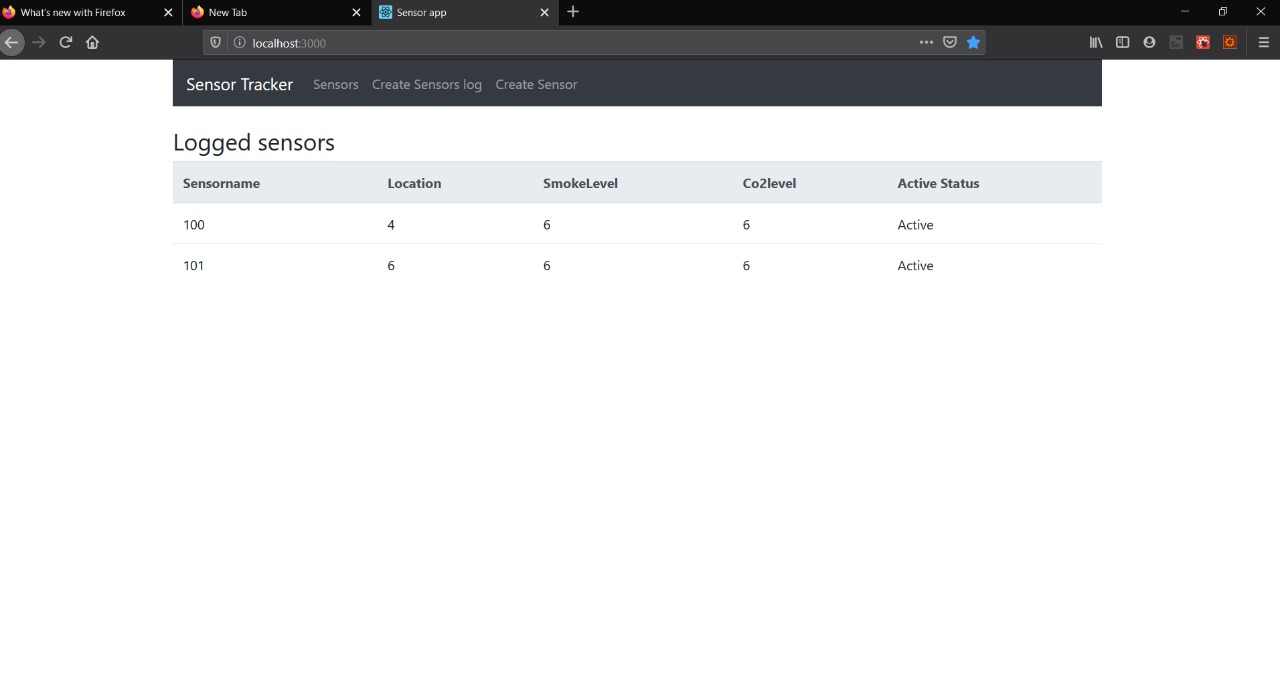
We used MYSQL work bench for the storage , the following diagram shows that the datas added in the database.



If the CO2 Level or Smoke Level exceeded 5, the alarm should be in ACTIVATED status else it should be DEACTIVATED. If Alarm activated also can get email notification, for the reference , the following diagram shows that email received,



Also the same details can viewed by the Desktop client app that was done by using REACT JS , the diagram shows the interface of the client app interface.



# Appendix

### REST API

const express=require('express');

const app=express();

const cors=require('cors');

const mongoose=require('mongoose');

require('dotenv').config();

const port=process.env.PORT || 5000;

//course middleware

app.use(cors());

app.use(express.json());

const uri=process.env.ATLAS\_URL;

mongoose.connect(uri,{useNewUrlParser:true,useCreateIndex:true,useUnifiedTopology:true});

const connection=mongoose.connection;

connection.once('open',() =>{

console.log("Mongo db database connection established successfully");

});

const sensorRouter=require('./routes/sensor');

const userRouter=require('./routes/users');

//someone enter exercise , it will load exerciseRoute

app.use('/sensors',sensorRouter);

app.use('/users',userRouter);

//start server

app.listen(port,() =>{

console.log('Server os running on port '+port)

});

sensor.model.js

const mongoose=require('mongoose');

const Schema=mongoose.Schema;

const sensorSchema = new Schema({

sensorname: { type: String, required: true},

location: {type: String, required: true},

smokelevel: {type: Number, required: true},

co2level: {type: Number, required:true},

activestatus: {type: Boolean,required:true}

},{

timestamp: false,

});

const Sensor=mongoose.model('Sensor',sensorSchema);

module.exports=Sensor;

user.model.js

const mongoose=require('mongoose');

const Schema=mongoose.Schema;

const userSchema=new Schema({

username:{

type: String,

required: true,

unique: true,

trim: true,

minlength: 3

},

}, {

timestamps:true,

});

const User= mongoose.model('Users',userSchema);

module.exports=User

sensor.js

const router=require('express').Router();

let Sensor=require('../models/sensor.model');

router.route('/').get((req,res) =>{

Sensor.find()

.then(sensors =>res.json(sensors))

.catch(err =>res.status(400).json('Error'+err))

});

router.route('/add').post((req,res) =>{

const sensorname=req.body.sensorname;

const location=req.body.location;

const smokelevel=req.body.smokelevel;

const co2level=req.body.co2level;

const activestatus=req.body.activestatus;

const newSensor=new Sensor({

sensorname,

location,

smokelevel,

co2level,

activestatus

});

newSensor.save()

.then(()=> res.json('Sensor added'))

.catch(err => res.status(400).json('Error'+err))

});

router.route('/:id').get((req,res) =>{

Sensor.findById(req.params.id)

.then(sensor => res.json(sensor))

.catch(err => res.json('Error' + err))

});

router.route('/:id').delete((req,res) =>{

Sensor.findByIdAndDelete(req.params.id)

.then(()=>res.json('Sensor deleted'))

.catch(err => res.status(400).json('Error'+err));

});

router.route('/update/:id').post((req,res) =>{

Sensor.findById(req.params.id)

.then(sensor =>{

sensor.sensorname=req.body.sensorname;

sensor.location=req.body.location;

sensor.smokelevel=req.body.smokelevel;

sensor.co2level=req.body.co2level;

sensor.activestatus=req.body.activestatus

sensor.save()

.then(() => res.json('Sensor updated'))

.catch(err => res.status(400).json('Error'+err));

})

.catch((err =>res.status(400).json('Error'+err)));

});

module.exports=router;

user.js

const express=require('express');

let User=require('../models/user.model');

const router=express.Router();

router.route('/').get((req,res) =>{

User.find()

.then(user => res.json(user))

.catch(err => res.status(400).json('Error' + err));

});

router.route('/add').post((req,res) =>{

const username = req.body.username;

const newUser=new User({username});

newUser.save()

.then(() => res.json('User added'))

.catch(err => res.status(400).json('Error'+ err));

});

module.exports=router;

create-sensor.component.js

import React,{Component} from "react";

import axios from 'axios';

export default class CreateSensor extends Component{

constructor(props) {

super(props);

this.onChangeSensorName=this.onChangeSensorName.bind(this);

this.onChangeLocation=this.onChangeLocation.bind(this);

this.onChangeSmokelevel=this.onChangeSmokelevel.bind(this);

this.onChangeCo2level=this.onChangeCo2level.bind(this);

this.onChangeActivestatus=this.onChangeActivestatus.bind(this);

this.onSubmit =this.onSubmit.bind(this);

this.state={

sensorname: '',

location:'',

smokelevel: 0,

co2level: 0,

activestatus: true,

users:[]

}

}

componentDidMount() {

axios.get('http://localhost:5000/users/')

.then(response =>{

if ( response.data.length > 0){

this.setState({

users:response.data.map(user =>user.username),

sensorname: response.data[0].username

})

}

})

}

onChangeSensorName(e){

this.setState({

sensorname: e.target.value

});

}

onChangeLocation(e){

this.setState({

location: e.target.value

});

}

onChangeSmokelevel(e){

this.setState({

smokelevel: e.target.value

});

}

onChangeCo2level(e){

this.setState({

co2level: e.target.value

});

}

onChangeActivestatus(e){

this.setState({

activestatus: e.target.value

});

}

onSubmit(e){

e.preventDefault();

const sensor= {

sensorname: this.state.sensorname,

location: this.state.location,

smokelevel: this.state.smokelevel,

co2level: this.state.co2level,

activestatus: this.state.activestatus

}

console.log(sensor);

axios.post('http://localhost:5000/sensors/add',sensor)

.then(res => console.log(res.data))

window.location= '/';

}

render() {

return (

<div>

<h3>Create new Sensor Log</h3>

<form onSubmit={this.onSubmit}>

<div className='form-group'>

<label >Sensorname:</label>

<select ref='userInput' required className='form-control' value={this.state.sensorname}

onChange={this.onChangeSensorName}>

{this.state.users.map(function (user) {

return <option

key={user} value={user}>{user}

</option>

})

}

</select>

<div className='form-group'>

<label>Location:</label>

<input type={'text'} required className='form-control'

value={this.state.location} onChange={this.onChangeLocation}/>

</div>

<div className='form-group'>

<label>Smoke Level</label>

<input type='text' className='form-control'

value={this.state.smokelevel} onChange={this.onChangeSmokelevel}/>

</div>

<div className='form-group'>

<label>Co2 Level</label>

<input type='text' className='form-control'

value={this.state.co2level} onChange={this.onChangeCo2level}/>

</div>

<div className='form-group'>

<label>Active status</label>

<input type='text' className='form-control'

value={this.state.activestatus} onChange={this.onChangeActivestatus}/>

</div>

<div className='form-group'>

<input type='submit' value='Create Sensor log' className='btn btn-primary'/>

</div>

</div>

</form>

</div>

);

}

}

sensor-list.component.js

**//Client App**

import React, { Component } from 'react';

import { Badge} from 'reactstrap';

import axios from 'axios';

const Sensor = props => (

<tr>

<td>{props.sensor.sensorname}</td>

<td>{props.sensor.location}</td>

<td>{props.sensor.smokelevel}</td>

<td>{props.sensor.smokelevel}</td>

<td>Active{props.sensor.activestatus}</td>

</tr>

);

export default class SensorList extends Component {

constructor(props) {

super(props);

this.state = {sensors: []};

}

componentDidMount() {

setInterval( () =>{

axios.get('http://localhost:5000/sensors/')

.then(response => {

this.setState({ sensors: response.data })

})

.catch((error) => {

console.log(error);

})

},4000)

}

sensorList() {

return this.state.sensors.map(currentsensor => {

return <Sensor sensor={currentsensor} key={currentsensor.\_id}/>;

})

}

render() {

return (

<div>

<h3>Logged sensors</h3>

<table className="table">

<thead className="thead-light">

<tr>

<th>Sensorname</th>

<th>Location</th>

<th>SmokeLevel</th>

<th>Co2level</th>

<th>Active Status</th>

</tr>

</thead>

<tbody>

{ this.sensorList() }

</tbody>

</table>

</div>

)

}

}

create-user.component.js

import React,{Component} from "react";

import axios from 'axios';

export default class CreateUser extends Component{

constructor(props) {

super(props);

this.onChangeUsername=this.onChangeUsername.bind(this);

this.onSubmit=this.onSubmit.bind(this);

this.state={

username:'',

}

}

onChangeUsername(e){

this.setState({

username:e.target.value

});

}

onSubmit(e){

e.preventDefault();

const user={

username: this.state.username,

};

console.log(user);

axios.post('http://localhost:5000/users/add',user)

.then(res => console.log(res.data));

this.setState({

username:''

})

}

render() {

return (

<div>

<h3>Create New Sensor</h3>

<form onSubmit={this.onSubmit}>

<div className='form-group'>

<label>Sensor name:</label>

<input type='text' required className='form-control'

value={this.state.username} onChange={this.onChangeUsername}/>

</div>

<div className='form-group'>

<input type='submit' value='Create User' className='btn btn-primary'/>

</div>

</form>

</div>

);

}

/\*

\* 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 adminlogin;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Sutharsan

\*/

public class LoginForm extends javax.swing.JFrame {

/\*\*

\* Creates new form LoginForm

\*/

public LoginForm() {

initComponents();

Toolkit toolkit = getToolkit();

Dimension size = toolkit.getScreenSize();

setLocation(size.width/2 - getWidth()/2 , size.height/2 - getHeight()/2);

}

/\*\*

\* 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">//GEN-BEGIN:initComponents

private void initComponents() {

jButton1 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

username = new javax.swing.JTextField();

password = new javax.swing.JPasswordField();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jButton1.setText("Login");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel3.setText("Password");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel2.setText("User Name");

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jLabel1.setText("Admin Login");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(254, Short.MAX\_VALUE)

.addComponent(jLabel1)

.addGap(251, 251, 251))

.addGroup(layout.createSequentialGroup()

.addGap(69, 69, 69)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(username)

.addComponent(password, javax.swing.GroupLayout.DEFAULT\_SIZE, 357, Short.MAX\_VALUE))

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 108, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addGap(66, 66, 66)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(username, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addComponent(password, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(88, 88, 88)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(169, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

boolean f = false;

try

{

Registry reg=LocateRegistry.getRegistry("127.0.0.1",1999);

LoginInterface i=(LoginInterface)reg.lookup("Login");

f=i.getLogin(username.getText(), password.getText());

if(f == true){

JOptionPane.showMessageDialog(null, "Login successfully");

new NewFireSensor().setVisible(true);

this.setVisible(false);

}

else{

JOptionPane.showMessageDialog(null, "user and password Do not matched");

username.setText("");

password.setText("");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}//GEN-LAST:event\_jButton1ActionPerformed

/\*\*

\* @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(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginForm.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 LoginForm().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPasswordField password;

private javax.swing.JTextField username;

// End of variables declaration//GEN-END:variables

}

**//NewFireSensor.java**

/\*

\* 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 adminlogin;

import adminlogin.EditEx;

import adminlogin.AddNew;

import java.awt.Dimension;

import java.awt.Toolkit;

/\*\*

\*

\* @author Sutharsan

\*/

public class NewFireSensor extends javax.swing.JFrame {

/\*\*

\* Creates new form NewFireSensor

\*/

public NewFireSensor() {

initComponents();

Toolkit toolkit = getToolkit();

Dimension size = toolkit.getScreenSize();

setLocation(size.width/2 - getWidth()/2 , size.height/2 - getHeight()/2);

}

/\*\*

\* 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">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jButton1.setText("Add New");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jButton2.setText("Edit Existing");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel1.setFont(new java.awt.Font("Teko SemiBold", 1, 48)); // NOI18N

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Fire Sensor");

jLabel1.setAlignmentX(0.5F);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(33, 33, 33)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 403, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 308, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 403, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(41, 41, 41))

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 102, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(145, 145, 145)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 128, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 126, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(509, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

new AddNew().setVisible(true);

this.setVisible(false);

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

new EditEx().setVisible(true);

this.setVisible(false);

}//GEN-LAST:event\_jButton2ActionPerformed

/\*\*

\* @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(NewFireSensor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewFireSensor.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 NewFireSensor().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

// End of variables declaration//GEN-END:variables

}

**//AddNew.java**

/\*

\* 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 adminlogin;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Sutharsan

\*/

public class AddNew extends javax.swing.JFrame {

/\*\*

\* Creates new form AddNew

\*/

public AddNew() {

initComponents();

Toolkit toolkit = getToolkit();

Dimension size = toolkit.getScreenSize();

setLocation(size.width/2 - getWidth()/2 , size.height/2 - getHeight()/2);

}

/\*\*

\* 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">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

floor = new javax.swing.JTextField();

room = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

sid = new javax.swing.JTextField();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel2.setText("Floor No :");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel3.setText("Room ID : ");

room.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

roomActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 48)); // NOI18N

jLabel4.setText("For the new sensor registration.");

jButton1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jButton1.setText("Register");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel5.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel5.setText("Sensor ID :");

jButton2.setText("Back");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(33, 33, 33)

.addComponent(jLabel3)

.addGap(29, 29, 29)

.addComponent(room, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(15, 15, 15)

.addComponent(jLabel4))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(40, 40, 40)

.addComponent(jButton1))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel5)

.addGap(41, 41, 41)

.addComponent(sid, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(41, 41, 41)

.addComponent(floor, javax.swing.GroupLayout.PREFERRED\_SIZE, 290, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 94, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(63, 63, 63))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(32, 32, 32)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(61, 61, 61)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(sid, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(30, 30, 30)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(floor, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(34, 34, 34)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addComponent(room, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(42, 42, 42)

.addComponent(jLabel4)

.addGap(55, 55, 55)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 53, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/firealarm","root","");

String sql = "insert into sensor values(?,?,?,0,0,false)";

PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1,sid.getText());

pst.setString(2, floor.getText());

pst.setString(3, room.getText());

pst.executeUpdate();

JOptionPane.showMessageDialog(null, "Sucessfully registered");

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void roomActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_roomActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_roomActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

new NewFireSensor().setVisible(true);

this.setVisible(false);

}//GEN-LAST:event\_jButton2ActionPerformed

/\*\*

\* @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(AddNew.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AddNew.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 AddNew().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JTextField floor;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JTextField room;

private javax.swing.JTextField sid;

// End of variables declaration//GEN-END:variables

}

**//EditEx.java**

/\*

\* 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 adminlogin;

import firealarm.client.FireMonitor;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.net.MalformedURLException;

import java.rmi.RemoteException;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Sutharsan

\*/

public class EditEx extends javax.swing.JFrame {

/\*\*

\* Creates new form EditEx

\*/

public EditEx() {

initComponents();

Toolkit toolkit = getToolkit();

Dimension size = toolkit.getScreenSize();

setLocation(size.width/2 - getWidth()/2 , size.height/2 - getHeight()/2);

}

public ArrayList<Database> datalist(){

ArrayList<Database> datalist =new ArrayList<>();

try{

Class.forName("com.mysql.jdbc.Driver");

java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/firealarm", "root" , "");

String query="SELECT \* FROM sensor ";

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(query);

Database database;

while(rs.next()){

database = new Database(rs.getInt("sensorID"),rs.getInt("floorNo"),rs.getInt("CO2Level"),rs.getInt("smokeLevel"),rs.getString("Room"),rs.getString("status"));

datalist.add(database);

}

}

catch(Exception e){

e.printStackTrace();

}

return datalist;

}

public void show\_database(){

ArrayList<Database> list=datalist();

DefaultTableModel model = (DefaultTableModel)table.getModel();

Object[] row=new Object[6];

for(int i=0; i< list.size();i++){

row[0]=list.get(i).getSid();

row[1]=list.get(i).getFloor();

row[2]=list.get(i).getRoom();

row[3]=list.get(i).getSmoke();

row[4]=list.get(i).getCo2();

row[5]=list.get(i).getStatus();

model.addRow(row);

model.fireTableDataChanged();

}

// table.setModel(model);

}

/\*\*

\* 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">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jLabel1.setText("Sensor Details");

table.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"ID", "Floor", "Room", "Smoke", "Co2", "Status"

}

));

jScrollPane2.setViewportView(table);

jButton1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jButton1.setText("Display");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Back");

jButton2.setActionCommand("Back");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(29, 29, 29)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 925, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(401, 401, 401)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 117, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(36, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(327, 327, 327)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 94, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(51, 51, 51))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(28, 28, 28)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(181, 181, 181)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX\_VALUE)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 152, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(86, 86, 86))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// try {

// FireMonitor fm = new FireMonitor();

// String[] args =null;

// fm.main(args);

// } catch (RemoteException ex) {

// Logger.getLogger(EditEx.class.getName()).log(Level.SEVERE, null, ex);

// } catch (MalformedURLException ex) {

// Logger.getLogger(EditEx.class.getName()).log(Level.SEVERE, null, ex);

// }

show\_database();

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

new NewFireSensor().setVisible(true);

this.setVisible(false);

}//GEN-LAST:event\_jButton2ActionPerformed

/\*\*

\* @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(EditEx.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(EditEx.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 EditEx().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable table;

// End of variables declaration//GEN-END:variables

}

**//LoginImplementation**

**/\***

**\* 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 adminlogin;**

**/\*\***

**\***

**\* @author Sutharsan**

**\*/**

**import java.rmi.\*;**

**import java.rmi.server.UnicastRemoteObject;**

**import adminlogin.LoginInterface;**

**public class LoginImplementation extends UnicastRemoteObject implements LoginInterface**

**{**

**public LoginImplementation() throws RemoteException**

**{**

**}**

**/\*\***

**\***

**\* @param username**

**\* @param password**

**\* @return**

**\* @throws RemoteException**

**\*/**

**@Override**

**public boolean getLogin(String username,String password) throws RemoteException**

**{**

**boolean found = false;**

**try**

**{**

**if(username.equals("admin") && password.equals("admin123"))**

**{**

**return found = true;**

**}**

**else**

**{**

**return found = false;**

**}**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**return found;**

**}**

**}**

**//LoginInterface.java**

/\*

\* 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 adminlogin;

/\*\*

\*

\* @author Sutharsan

\*/

import java.rmi.\*;

public interface LoginInterface extends Remote

{

public boolean getLogin(String username,String password) throws RemoteException;

}

**//FireMonitor.java**

/\*

\* 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 firealarm.client;

import firealarm.server.FireSensorService;

import firealarm.server.Implement;

import java.net.MalformedURLException;

import java.rmi.NotBoundException;

import java.rmi.RemoteException;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.rmi.server.UnicastRemoteObject;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Random;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Sutharsan

\*/

public class FireMonitor extends UnicastRemoteObject implements FireSensorClient,Runnable{

private static final long serialVersionUID = 1L;

private int count = 0;

static int readingsmoke;

int newSmoke= 0;

int newCo2 = 0;

String activity ;

public FireMonitor() throws RemoteException{

super();

}

public static void main(String[] args) throws MalformedURLException, RemoteException{

System.setProperty("java.security.policy", "file:allowall.policy");

// Implement implement = new Implement();

// implement.updateSensor();

ArrayList<Integer> datalist =new ArrayList<>();

try{

Class.forName("com.mysql.jdbc.Driver");

java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/firealarm", "root" , "");

String query="SELECT sensorID FROM firealarm.sensor";

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(query);

// System.out.println(rs);

//

while(rs.next()) {

// System.out.println(rs.getInt("sensorID"));ta,

datalist.add(rs.getInt("sensorID"));

}

//

}

catch(Exception e){

e.printStackTrace();

}

try{

Registry reg=LocateRegistry.getRegistry("127.0.0.1",2000);

FireSensorService sensor=(FireSensorService)reg.lookup("FireServer");

for(int i = 0 ; i <= datalist.size(); i++){

System.out.println("SensorID :"+ datalist.get(i));

readingsmoke = sensor.getSmokeLevel();

System.out.println("Smoke Reading : " + readingsmoke);

int readingCO2 = sensor.getCO2Level();

System.out.println("CO2 Reading : " + readingCO2);

int readingfloor = sensor.getFloorNumber();

System.out.println("Floor : " + readingfloor);

int readingroom = sensor.getRoomNumber();

System.out.println("Room : " + readingroom);

boolean readingactiviy = sensor.getActivity();

System.out.println("Status : " + readingactiviy);

//A client object created

// FireMonitor monitor = new FireMonitor();

sensor.addSensor( new FireMonitor());

new FireMonitor().run();

}

} catch (RemoteException re) {

System.out.println(re);

} catch (NotBoundException nbe) {

System.out.println(nbe);

}

}

@Override

public void run(){

for(;;){

try{

Thread.sleep(1000);

this.smokechange(count);

this.CO2change(count);

this.activitychange(true);

this.updateSensor();

}

catch(InterruptedException ie){

System.out.println(ie);

} catch (RemoteException ex) {

Logger.getLogger(FireMonitor.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

@Override

public void smokechange(int smoke) throws RemoteException{

Random random = new Random();

newSmoke = random.nextInt(11);

if(newSmoke >= 5){

SendMailSSL se = new SendMailSSL();

String[] ab = null;

se.main(ab);

System.err.println("\nSmoke change event : " +newSmoke);

activity = "Activated";

}

else{

System.out.println("\nCO2 change event : " + newSmoke);

activity = "Deactivated";

}

}

@Override

public void CO2change(int CO2) throws RemoteException{

Random random = new Random();

newCo2 = random.nextInt(11);

if(newCo2 >= 5 ){

System.err.println("\nCO2 change event : " + newCo2);

activity = "Activated";

}

else{

System.out.println("\nCO2 change event : " + newCo2);

activity = "Deactivated";

}

}

@Override

public void activitychange(boolean activity) throws RemoteException {

System.out.println("\nActivity change event : " + activity);

}

@Override

public void updateSensor() throws RemoteException {

try{

Class.forName("com.mysql.jdbc.Driver");

java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/firealarm", "root" , "");

String query = "update sensor set smokeLevel = ? , CO2Level = ?, status = ? where sensorID = ?";

PreparedStatement preparedStmt = con.prepareStatement(query);

preparedStmt.setInt(1,newSmoke );

preparedStmt.setInt(2,newCo2 );

preparedStmt.setString(3,activity );

preparedStmt.setInt(4,101);

// execute the java preparedstatement

preparedStmt.executeUpdate();

}

catch(Exception e){

e.printStackTrace();

}

}

}

**//FireSensorClient**

/\*

\* 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 firealarm.client;

/\*\*

\*

\* @author Sutharsan

\*/

import java.rmi.RemoteException;

public interface FireSensorClient extends java.rmi.Remote

{

public void smokechange(int smoke) throws RemoteException;

public void CO2change(int CO2) throws RemoteException;

public void activitychange(boolean activity) throws RemoteException;

public void updateSensor() throws RemoteException;

}

**//FireAlarmServer**

/\*

\* 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 firealarm.server;

/\*\*

\*

\* @author Sutharsan

\*/

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import adminlogin.LoginImplementation;

import java.rmi.\*;

import java.rmi.server.UnicastRemoteObject;

import adminlogin.LoginInterface;

public class FireAlarmServer extends UnicastRemoteObject

{

private static final long serialVersionUID = 1L;

public FireAlarmServer() throws RemoteException{

super();

}

public static void main(String[]args)

{

try

{

Registry reg = LocateRegistry.createRegistry(2999);

LoginInterface li = new LoginImplementation();

reg.rebind("Login", li);

}

catch(Exception e)

{

e.printStackTrace();

}

System.setProperty("java.security.policy","file:allowall.policy");

try

{

Registry reg = LocateRegistry.createRegistry(3999);

FireSensorService fs = new Implement();

reg.rebind("FireServer", fs);

System.err.println("Loading Sensor server....");

System.err.println("Server is ready");

Thread thread = new Thread((Runnable) fs);

thread.start();

}

catch(Exception e)

{

System.out.println("Error - " + e);

}

}

}

**//FireSensorService**

/\*

\* 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 firealarm.server;

import firealarm.client.FireSensorClient;

/\*\*

\*

\* @author Sutharsan

\*/

public interface FireSensorService extends java.rmi.Remote

{

//register sensor

public void addSensor(FireSensorClient fireclient) throws java.rmi.RemoteException;

//smoke level returned from client

public int getSmokeLevel() throws java.rmi.RemoteException;

//CO2 level returned from client

public int getCO2Level() throws java.rmi.RemoteException;

//floor level returned from client

public int getFloorNumber() throws java.rmi.RemoteException;

//room level returned from client

public int getRoomNumber() throws java.rmi.RemoteException;

public boolean getActivity() throws java.rmi.RemoteException;

}

**//Implement**

/\*

\* 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 firealarm.server;

import firealarm.client.FireSensorClient;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import java.util.Random;

/\*\*

\*

\* @author Sutharsan

\*/

public class Implement extends UnicastRemoteObject implements FireSensorService, Runnable {

public Implement() throws RemoteException{

super();

//// this.smoke = a.nextInt(10)is

this.smoke = new Random().nextInt(10);

this.CO2 = new Random().nextInt(10);

// this.CO2 = b.nextInt(10);

}

private static final long serialVersionUID = 1L;

public volatile int smoke;

public volatile int CO2;

public int floor;

public int room;

public boolean activity;

private ArrayList<FireSensorClient> list = new ArrayList<FireSensorClient>();

@Override

public void run()

{

Random r = new Random();

for(;;)

{

try

{

//Sleep for a random amount of time

int duration = r.nextInt() % 10000 + 200;

//Check to see if negative, if so, reverse

if(duration < 0)

{

duration = duration \* -1;

Thread.sleep(duration);

}

}

catch(InterruptedException ie)

{

System.out.println(ie);

}

//Notify registered listeners

notifySensor();

}

}

private void notifySensor()

{

for(FireSensorClient fireclient:list)

{

try

{

fireclient.smokechange(smoke);

fireclient.CO2change(CO2);

fireclient.activitychange(activity);

}

catch(RemoteException e)

{

e.printStackTrace();

}

}

}

@Override

public void addSensor(FireSensorClient fireclient) throws RemoteException

{

System.out.println("Adding Sensor - " + fireclient);

list.add(fireclient);

}

@Override

public int getSmokeLevel() throws RemoteException

{

return smoke;

}

@Override

public int getCO2Level() throws RemoteException

{

return CO2;

}

@Override

public int getFloorNumber() throws RemoteException

{

return floor;

}

@Override

public int getRoomNumber() throws RemoteException

{

return room;

}

@Override

public boolean getActivity() throws RemoteException {

return true;

}

}

**//SendMailSSL.java**

/\*

\* 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 firealarm.client;

/\*\*

\*

\* @author Sutharsan

\*/

import java.util.Properties;

import javax.mail.\*;

import javax.mail.internet.\*;

class Mailer{

public static void send(String from,String password,String to,String sub,String msg){

//Get properties object

Properties props = new Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class",

"javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

//get Session

Session session = Session.getDefaultInstance(props,

new javax.mail.Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(from,password);

}

});

//compose message

try {

MimeMessage message = new MimeMessage(session);

message.addRecipient(Message.RecipientType.TO,new InternetAddress(to));

message.setSubject(sub);

message.setText(msg);

//send message

Transport.send(message);

System.out.println("message sent successfully");

} catch (MessagingException e) {throw new RuntimeException(e);}

}

}

public class SendMailSSL{

public static void main(String[] args) {

//from,password,to,subject,message

Mailer.send("sliitsutharshan@gmail.com","Pavalam62","sliitsutharshan@gmail.com"," FIRE ALARM ALERT!!","There is a allert on the Fire Sensor it may me Dangerous Quickly go and Check! ");

//change from, password and to

}

}