

PAT (Practical Assessment Task) – Grade 11: Phase 4

Name

Semira Nee-Whang

GUI: Filter
and sorting
of data
(screenshot)

WaterManagement

CapacityML:

AllocatedML: Source Name Start with: Sort by:

UsedML:

Source ID	Source Name	Source Type	Province	Sector	CapacityML	AllocatedML	UsedML	Date Recorded	IsActive
1	Orange River	River	Northern Cape	Agriculture	120000	40000	38000	2025-03-10	TRUE
2	uThukela River	River	KwaZulu-Natal	Domestic	100000	20000	21000	2025-03-12	TRUE
3	Bloemhof Dam	Dam	North West / Free State	Industry	95000	15000	12000	2025-03-18	TRUE
4	Vaal River	River	Gauteng	Domestic	150000	30000	29000	2025-03-25	TRUE
5	Gariiep Dam	Dam	Eastern Cape	Agriculture	130000	25000	26000	2025-03-28	TRUE
6	Limpopo Borehole	Borehole	Limpopo	Domestic	50000	10000	8000	2025-04-01	FALSE
7	Inyaka Dam	Dam	Mpumalanga	Agriculture	110000	30000	27000	2025-04-03	TRUE
8	Hartebeespoort	Dam	North West	Industry	87000	18000	17500	2025-04-05	TRUE
9	Umgeni River	River	KwaZulu-Natal	Domestic	92000	22000	21500	2025-04-07	TRUE
10	Claanwilliam Dam	Dam	Western Cape	Agriculture	98000	26000	24000	2025-04-10	TRUE

GUI: Insert
(screenshot)

Source Name:

Type:

Province:

Sector:

Total Capacity:

Allocated Capacity:

Used Capacity:

Date Recorded: Format: YYYY - MM - DD

Is Active:

GUI: Delete
(screenshot)

ID of the product you want to delete:

Project	Upload complete project as a zip file
Code	DB Class <pre> import java.sql.*; import javax.swing.*; public class DB { //Declare properties private final String driver = "net.ucanaccess.jdbc.UcanaccessDriver"; //DB must be in project folder private final String url = "jdbc:ucanaccess://D:/Documents/NetBeansProjects/PAT/WaterDB.accdb"; private Connection connection; private Statement statement; private ResultSet resultSet; //Constructor method public DB() { //Load driver try { Class.forName(driver); System.out.println("Driver found"); } catch (ClassNotFoundException e) //Trap the error if the driver is not found { JOptionPane.showMessageDialog(null, "Error: Database Driver not found"); } //Create connection try { connection = DriverManager.getConnection(url); JOptionPane.showMessageDialog(null, "Connected successfully!"); } catch (SQLException e) { JOptionPane.showMessageDialog(null, "Unable to connect: " + e.getMessage()); e.printStackTrace(); // print detailed error in console } } public ResultSet queryDB(String inStatement) throws SQLException //This is the generic query to execute a SELECT statement { //Query the database statement = connection.createStatement(); System.out.println(inStatement); resultSet = statement.executeQuery(inStatement); //Return data as a resultset return resultSet; } public void changeDB (String inStatement) throws SQLException //This is the generic query to execute an INSERT/ UPDATE/ DELETE STATEMENT { System.out.println(inStatement); statement = connection.createStatement(); //Execute the SQL Statement statement.executeUpdate(inStatement); statement.close(); } } </pre>

FrmWatermanagement Class

```
import javax.swing.JOptionPane;

public class FrmWatermanagemnt extends javax.swing.JFrame {

    private static final java.util.logging.Logger logger =
        java.util.logging.Logger.getLogger(FrmWatermanagemnt.class.getName());
    WaterManagementManager waterManagement = new WaterManagementManager();

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

    private void populateWaterData() {

        //This code shows the water data on the screen
        //clear the text area
        txtaWater.setText(" ");

        //declare variables that will represent WHERE and ORDER BY clauses
        //of the Select statement
        String where = "";
        String sort = "SourceID"; //default

        //Check if all data should be shown
        //If values are set, create a WHERE clause
        if (!cmbCapacityML.getSelectedItem().equals("All"))
        {
            where = "CapacityML " + cmbCapacityML.getSelectedItem();
        }

        if (!cmbAllocatedML.getSelectedItem().equals("All")){
            where = "AllocatedML " + cmbAllocatedML.getSelectedItem();
        }

        if (!cmbUsedML.getSelectedItem().equals("All")){
            where = "UsedML " + cmbUsedML.getSelectedItem();
        }

        //If the "Start with" textfield is filled in, add to / Create a WHERE
        //clause
        if (!txtSourceNameStartWith.getText().isEmpty()) {

            // If WHERE clause is still empty, create WHERE clause
            if (where.isEmpty()) {
                where = "SourceName LIKE '" + txtSourceNameStartWith.getText() +
                    "'*";
            }
        } else { // If WHERE clause is populated, add to it
            where += " AND SourceName LIKE '" + txtSourceNameStartWith.getText()
                + "'*";
        }

        switch(cmbSortBy.getSelectedItem().toString())
        {
            case "Source ID" :
                sort = "SourceID";
                break;
            case "Source Name":
                sort = "SourceName";
                break;
            case "Sector":
                sort = "Sector";
                break;
            case "Capacity ML":
                sort = "CapacityML";
                break;
            case "Allocated ML":
                sort = "AllocatedML";
            }
    }
}
```

```

        break;
    case "Used ML":
        sort = "UsedML";
        break;
    case "Date Recorded":
        sort = "DateRecorded";
        break;
    case "Is Active":
        sort = "IsActive";
        break;

    default:
        //Province
        //Source Type
        sort = cmbSortBy.getSelectedItem().toString();
    }

    //Fill the text area with data
    txtaWater.setText(waterManagement.getWaterData(where, sort));
}

/**
 * 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() {

    frmAdd = new javax.swing.JFrame();
    lblSourceNameOnAddForm = new javax.swing.JLabel();
    txtSourceNameOnAddForm = new javax.swing.JTextField();
    lblTypeOnAddForm = new javax.swing.JLabel();
    txtTypeOnAddForm = new javax.swing.JTextField();
    lblProvinceOnAddForm = new javax.swing.JLabel();
    txtProvinceOnAddForm = new javax.swing.JTextField();
    txtSectorOnAddForm = new javax.swing.JTextField();
    lblSectorOnAddForm = new javax.swing.JLabel();
    txtCapacityOnAddForm = new javax.swing.JTextField();
    lblCapacityOnAddForm = new javax.swing.JLabel();
    txtAllocatedCapacityOnAddForm = new javax.swing.JTextField();
    lblAllocatedCapacityOnAddForm = new javax.swing.JLabel();
    txtUsedCapacityOnAddForm = new javax.swing.JTextField();
    lblUsedCapacityOnAddForm = new javax.swing.JLabel();
    txtDateRecordedOnAddForm = new javax.swing.JTextField();
    lblDateRecordedOnAddForm = new javax.swing.JLabel();
    lblFormatOnAddForm = new javax.swing.JLabel();
    lblIsActiveOnAddForm = new javax.swing.JLabel();
    btnAddOnAddForm = new javax.swing.JButton();
    btnCancelOnAddForm = new javax.swing.JButton();
    cmbIsactiveOnAddForm = new javax.swing.JComboBox<>();
    frmDelete = new javax.swing.JFrame();
    lblIDdeleteOnDELETEForm = new javax.swing.JLabel();
    txtIDdeleteOnDELETEForm = new javax.swing.JTextField();
    btnDeleteOnDELETEForm = new javax.swing.JButton();
    btnCancelOnDELETEForm = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    txtaWater = new javax.swing.JTextArea();
    jLabel1 = new javax.swing.JLabel();
    lblSourceNameStartWith = new javax.swing.JLabel();
    cmbCapacityML = new javax.swing.JComboBox<>();
    btnSearch = new javax.swing.JButton();
    lblCapacityML = new javax.swing.JLabel();
    lblSortBy = new javax.swing.JLabel();
    cmbSortBy = new javax.swing.JComboBox<>();
    txtSourceNameStartWith = new javax.swing.JTextField();
    lblUsedML = new javax.swing.JLabel();
    cmbUsedML = new javax.swing.JComboBox<>();
    lblAllocatedML = new javax.swing.JLabel();
    cmbAllocatedML = new javax.swing.JComboBox<>();
    btnAdd = new javax.swing.JButton();
    btnDelete = new javax.swing.JButton();

    frmAdd.setMinimumSize(new java.awt.Dimension(900, 600));

```

```

lblSourceNameOnAddForm.setText("Source Name:");

txtSourceNameOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSourceNameOnAddFormActionPerformed(evt);
    }
});

lblTypeOnAddForm.setText("Type:");

txtTypeOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtTypeOnAddFormActionPerformed(evt);
    }
});

lblProvinceOnAddForm.setText("Province:");

txtProvinceOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtProvinceOnAddFormActionPerformed(evt);
    }
});

txtSectorOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSectorOnAddFormActionPerformed(evt);
    }
});

lblSectorOnAddForm.setText("Sector:");

txtCapacityOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtCapacityOnAddFormActionPerformed(evt);
    }
});

lblCapacityOnAddForm.setText("Total Capacity:");

txtAllocatedCapacityOnAddForm.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAllocatedCapacityOnAddFormActionPerformed(evt);
    }
});

lblAllocatedCapacityOnAddForm.setText("Allocated Capacity:");

txtUsedCapacityOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtUsedCapacityOnAddFormActionPerformed(evt);
    }
});

lblUsedCapacityOnAddForm.setText("Used Capacity:");

txtDateRecordedOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtDateRecordedOnAddFormActionPerformed(evt);
    }
});

lblDateRecordedOnAddForm.setText("Date Recorded:");

lblFormatOnAddForm.setText("Format: YYYY - MM - DD");

lblIsActiveOnAddForm.setText("Is Active:");

btnAddOnAddForm.setText("Add");
btnAddOnAddForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddOnAddFormActionPerformed(evt);
    }
});

```



```

javax.swing.GroupLayout.PREFERRED_SIZE, 138, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(cmbIsActiveOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(51, Short.MAX_VALUE))
    );
    frmAddLayout.setVerticalGroup(
        frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(frmAddLayout.createSequentialGroup())
        .addGap(35, 35, 35)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblSourceNameOnAddForm)
        .addComponent(txtSourceNameOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtTypeOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblTypeOnAddForm))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblProvinceOnAddForm)
        .addComponent(txtProvinceOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblSectorOnAddForm)
        .addComponent(txtSectorOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblCapacityOnAddForm)
        .addComponent(txtCapacityOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblAllocatedCapacityOnAddForm)
        .addComponent(txtAllocatedCapacityOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblUsedCapacityOnAddForm)
        .addComponent(txtUsedCapacityOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblDateRecordedOnAddForm)
        .addComponent(txtDateRecordedOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblFormatOnAddForm))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblIsActiveOnAddForm)
        .addComponent(cmbIsActiveOnAddForm,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17,
Short.MAX_VALUE)

```

```

.addGroup(frmAddLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnAddOnAddForm)
    .addComponent(btnCancelOnAddForm))
    .addGap(35, 35, 35))
);

frmDelete.setMinimumSize(new java.awt.Dimension(333, 210));

lblIDdeleteOnDELETEDForm.setText("ID of the product you want to delete:");

btnDeleteOnDELETEDForm.setText("Delete");
btnDeleteOnDELETEDForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnDeleteOnDELETEDFormActionPerformed(evt);
    }
});

btnCancelOnDELETEDForm.setText("Cancel");
btnCancelOnDELETEDForm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelOnDELETEDFormActionPerformed(evt);
    }
});

javax.swing.GroupLayout frmDeleteLayout = new
javax.swing.GroupLayout(frmDelete.getContentPane());
frmDelete.getContentPane().setLayout(frmDeleteLayout);
frmDeleteLayout.setHorizontalGroup(

frmDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(frmDeleteLayout.createSequentialGroup()
        .addGap(24, 24, 24)

.addGroup(frmDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(frmDeleteLayout.createSequentialGroup()
        .addGroup(frmDeleteLayout.createSequentialGroup()
            .addComponent(btnDeleteOnDELETEDForm)
            .addGap(37, 37, 37)
            .addComponent(btnCancelOnDELETEDForm))
        .addComponent(lblIDdeleteOnDELETEDForm,
            javax.swing.GroupLayout.PREFERRED_SIZE, 217, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtIDdeleteOnDELETEDForm,
            javax.swing.GroupLayout.PREFERRED_SIZE, 35, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(51, Short.MAX_VALUE))
    );
frmDeleteLayout.setVerticalGroup(

frmDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(frmDeleteLayout.createSequentialGroup()
        .addGap(57, 57, 57)

.addGroup(frmDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblIDdeleteOnDELETEDForm)
    .addComponent(txtIDdeleteOnDELETEDForm,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(45, 45, 45)

.addGroup(frmDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnDeleteOnDELETEDForm)
    .addComponent(btnCancelOnDELETEDForm))
        .addContainerGap(63, Short.MAX_VALUE))
    );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("WaterManagement");

txaWater.setColumns(20);
txaWater.setFont(new java.awt.Font("Monospaced", 0, 12)); // NOI18N
txaWater.setRows(5);
jScrollPane1.setViewportView(txWater);

jLabel1.setText("jLabel1");

lblSourceNameStartWith.setText("Source Name Start with:");

```



```

        cmbCapacityML.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
            "All", "<10000", "<20000", "<30000", "<40000", "\t" }));
        cmbCapacityML.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbCapacityMLActionPerformed(evt);
            }
        });

        btnSearch.setText("Search");
        btnSearch.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnSearchActionPerformed(evt);
            }
        });

        lblCapacityML.setText("CapacityML:");

        lblSortBy.setText("Sort by:");

        cmbSortBy.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Source
ID", "Source Name", "Source Type", "Province", "Sector", "Capacity", "AllocatedML",
"UsedML", "DateRecorded", "IsActive" }));
        cmbSortBy.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbSortByActionPerformed(evt);
            }
        });

        txtSourceNameStartWith.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtSourceNameStartWithActionPerformed(evt);
            }
        });

        lblUsedML.setText("UsedML:");

        cmbUsedML.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "All",
"<10000", "<20000", "<30000", "<40000", "\t" }));
        cmbUsedML.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbUsedMLActionPerformed(evt);
            }
        });

        lblAllocatedML.setText("AllocatedML:");

        cmbAllocatedML.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
            "All", "<10000", "<20000", "<30000", "<40000", "\t" }));
        cmbAllocatedML.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmbAllocatedMLActionPerformed(evt);
            }
        });

        btnAdd.setText("Add");
        btnAdd.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnAddActionPerformed(evt);
            }
        });

        btnDelete.setText("Delete");
        btnDelete.setMinimumSize(new java.awt.Dimension(75, 30));
        btnDelete.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnDeleteActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(lblCapacityML)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(cmbCapacityML, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addComponent(lblSortBy)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(cmbSortBy, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addComponent(lblUsedML)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(cmbUsedML, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addComponent(lblAllocatedML)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(cmbAllocatedML, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addComponent(btnAdd)
                            .addGap(10, 10, 10)
                            .addComponent(btnDelete)
                        )
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(txtSourceNameStartWith, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addComponent(btnSearch)
                        )
                    )
                )
        );
    }

    private void cmbCapacityMLActionPerformed(java.awt.event.ActionEvent evt) {}

    private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {}

    private void cmbSortByActionPerformed(java.awt.event.ActionEvent evt) {}

    private void txtSourceNameStartWithActionPerformed(java.awt.event.ActionEvent evt) {}

    private void cmbUsedMLActionPerformed(java.awt.event.ActionEvent evt) {}

    private void cmbAllocatedMLActionPerformed(java.awt.event.ActionEvent evt) {}

    private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {}

    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {}
}

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jScrollPane1)
    .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addComponent(lblUsedML,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(lblCapacityML,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(lblAllocatedML,
javax.swing.GroupLayout.DEFAULT_SIZE, 81, Short.MAX_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
        .addGap(218, 218, 218)
        .addComponent(lblSourceNameStartWith)
        .addGap(18, 18, 18)
        .addComponent(txtSourceNameStartWith,
javax.swing.GroupLayout.PREFERRED_SIZE, 83, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(85, 85, 85)
        .addComponent(lblSortBy)
        .addGap(18, 18, 18)
        .addComponent(cmbSortBy,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnSearch))
    .addGroup(layout.createSequentialGroup())
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(cmbAllocatedML,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(cmbCapacityML,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(cmbUsedML,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(0, 816, Short.MAX_VALUE)))
    .addGroup(layout.createSequentialGroup())
        .addComponent(btnAdd)
        .addGap(51, 51, 51)
        .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    .addContainerGap())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addGap(0, 0, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblCapacityML, javax.swing.GroupLayout.PREFERRED_SIZE,
22, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(cmbCapacityML, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblAllocatedML, javax.swing.GroupLayout.PREFERRED_SIZE,
22, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(cmbAllocatedML, javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(lblSourceNameStartWith,
javax.swing.GroupLayout.PREFERRED_SIZE, 22, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtSourceNameStartWith,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(lblSortBy, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cmbSortBy, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnSearch))
        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(lblUsedML, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cmbUsedML, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25,
Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 315,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnAdd)
            .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(31, 31, 31))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 259, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(0, 260, Short.MAX_VALUE)))
        );

        setSize(new java.awt.Dimension(1015, 543));
        setLocationRelativeTo(null);
    } // </editor-fold> // GEN-END: initComponents

    private void cmbCapacityMLActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbCapacityMLActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_cmbCapacityMLActionPerformed

    private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_btnSearchActionPerformed
        populateWaterData();
    } // GEN-LAST:event_btnSearchActionPerformed

    private void cmbSortByActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbSortByActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_cmbSortByActionPerformed

    private void txtSourceNameStartWithActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_txtSourceNameStartWithActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_txtSourceNameStartWithActionPerformed

    private void cmbUsedMLActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbUsedMLActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_cmbUsedMLActionPerformed

    private void cmbAllocatedMLActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_cmbAllocatedMLActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_cmbAllocatedMLActionPerformed

    private void txtSourceNameOnAddFormActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_txtSourceNameOnAddFormActionPerformed
        // TODO add your handling code here:
    } // GEN-LAST:event_txtSourceNameOnAddFormActionPerformed

```

```

private void txtTypeOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtTypeOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtTypeOnAddFormActionPerformed

private void txtProvinceOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtProvinceOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtProvinceOnAddFormActionPerformed

private void txtSectorOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtSectorOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtSectorOnAddFormActionPerformed

private void txtCapacityOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtCapacityOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtCapacityOnAddFormActionPerformed

private void txtAllocatedCapacityOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtAllocatedCapacityOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtAllocatedCapacityOnAddFormActionPerformed

private void txtUsedCapacityOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtUsedCapacityOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtUsedCapacityOnAddFormActionPerformed

private void txtDateRecordedOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_txtDateRecordedOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_txtDateRecordedOnAddFormActionPerformed

private void cmbIsactiveOnAddFormItemStateChanged(java.awt.event.ItemEvent evt) { //GEN-FIRST:event_cmbIsactiveOnAddFormItemStateChanged
    // TODO add your handling code here:
} //GEN-LAST:event_cmbIsactiveOnAddFormItemStateChanged

private void cmbIsactiveOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cmbIsactiveOnAddFormActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_cmbIsactiveOnAddFormActionPerformed

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnAddActionPerformed

    //When the Add button is selected, show popup
    //This will centre the sub form
    frmAdd.setLocationRelativeTo(this);
    //initialise components
    txtSourceNameOnAddForm.setText("");
    txtTypeOnAddForm.setText("");
    txtProvinceOnAddForm.setText("");
    txtSectorOnAddForm.setText("");
    txtCapacityOnAddForm.setText("");
    txtAllocatedCapacityOnAddForm.setText("");
    txtUsedCapacityOnAddForm.setText("");
    txtDateRecordedOnAddForm.setText("");
    cmbIsactiveOnAddForm.setSelectedIndex(0);

    //Show the sub-form
    frmAdd.setVisible(true);
} //GEN-LAST:event_btnAddActionPerformed

private void btnCancelOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnCancelOnAddFormActionPerformed
    //Hide form when cancel button is selected
    frmAdd.setVisible(false);
} //GEN-LAST:event_btnCancelOnAddFormActionPerformed

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_btnDeleteActionPerformed
    //When the Delete button is selected, show popup form
    //This will centre the sub-form

```

```

        frmDelete.setLocationRelativeTo(this);
        //initialise components
        txtlDdeleteOnDELETEDForm.setText("");
        //show the sub-form
        frmDelete.setVisible(true);
    } //GEN-LAST:event_btnDeleteActionPerformed

    private void btnCancelOnDELETEDFormActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_btnCancelOnDELETEDFormActionPerformed
        //Hide form when the Cancel button is selected
        frmDelete.setVisible(false);
    } //GEN-LAST:event_btnCancelOnDELETEDFormActionPerformed

    private void btnDeleteOnDELETEDFormActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_btnDeleteOnDELETEDFormActionPerformed
        //When the delete button on the sub form is selected check if data valid and
        delete the product
        // Check if ID is filled in
        if (txtlDdeleteOnDELETEDForm.getText().isEmpty()) {
            //Display error meesage if ID not supplied
            JOptionPane.showMessageDialog(null, "Please fill in the ID");
        }
        else
        {
            try
            {
                //Check if Id is numeric. The next ststatement will fail if not
                //the "catch" part will display an error
                int num = Integer.parseInt(txtlDdeleteOnDELETEDForm.getText());

                //Comfim deletion
                int answer = JOptionPane.showConfirmDialog(null, "Are you sure"
                    + "you want to delete this product", "Confirm",
                    JOptionPane.YES_NO_OPTION);
                if (answer == 0) // yes
                {
                    waterManagement.deleteWaterDataFromDB(Integer.parseInt(
                        txtlDdeleteOnDELETEDForm.getText()));
                    //Refresh Source info
                    populateWaterData();
                }
                //Hide form
                frmDelete.setVisible(false);
            }
            catch (NumberFormatException i)
            { // Trap error of non numeric ID is entered
                {
                    //Display error message if ID is not numeric
                    JOptionPane.showMessageDialog(null,
                        "Please fill in a numeric value for the ID");
                }
            }
        }
    } //GEN-LAST:event_btnDeleteOnDELETEDFormActionPerformed

    private void btnAddOnAddFormActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_btnAddOnAddFormActionPerformed
        // Check if data is filled in and valid, then add the record
        if (txtSourceNameOnAddForm.getText().isEmpty() ||
            txtTypeOnAddForm.getText().isEmpty() ||
            txtProvinceOnAddForm.getText().isEmpty() ||
            txtSectorOnAddForm.getText().isEmpty() ||
            txtCapacityOnAddForm.getText().isEmpty() ||
            txtAllocatedCapacityOnAddForm.getText().isEmpty() ||
            txtUsedCapacityOnAddForm.getText().isEmpty() ||
            txtDateRecordedOnAddForm.getText().isEmpty())
        {
            // Display error message if all values are not supplied
            JOptionPane.showMessageDialog(null, "Please fill in all values!");
        }
        else
        {
            // Insert the new water data record
            // The Source ID is likely auto-generated in the database, so it's not included here.
            // Note: The data types are assumed based on your fields (int for ML fields, String
            for others).
            // You may need to add error handling for Integer.parseInt and Double.parseDouble.

```

```

        waterManagement.addWaterDataToDB(
            txtSourceNameOnAddForm.getText(),
            txtTypeOnAddForm.getText(),
            txtProvinceOnAddForm.getText(),
            txtSectorOnAddForm.getText(),
            Integer.parseInt(txtCapacityOnAddForm.getText()),
            Integer.parseInt(txtAllocatedCapacityOnAddForm.getText()),
            Integer.parseInt(txtUsedCapacityOnAddForm.getText()),
            txtDateRecordedOnAddForm.getText(), // Date/Time fields often need careful
formatting
            cmbIsactiveOnAddForm.getSelectedItem().toString() // Get the selected "Yes/No"
value
        );

        // Populate the main table with the new data
        populateWaterData();

        // Hide the add form
        frmAdd.setVisible(false);
    }

    //GEN-LAST:event_btnAddOnAddFormActionPerformed

    /**
     * @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 (ReflectiveOperationException |
            javax.swing.UnsupportedLookAndFeelException ex) {
            logger.log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(() -> new FrmWatermanagemnt()
            .setVisible(true));
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton btnAdd;
    private javax.swing.JButton btnAddOnAddForm;
    private javax.swing.JButton btnCancelOnAddForm;
    private javax.swing.JButton btnCancelOnDELETEForm;
    private javax.swing.JButton btnDelete;
    private javax.swing.JButton btnDeleteOnDELETEForm;
    private javax.swing.JButton btnSearch;
    private javax.swing.JComboBox<String> cmbAllocatedML;
    private javax.swing.JComboBox<String> cmbCapacityML;
    private javax.swing.JComboBox<String> cmbIsactiveOnAddForm;
    private javax.swing.JComboBox<String> cmbSortBy;
    private javax.swing.JComboBox<String> cmbUsedML;
    private javax.swing.JFrame frmAdd;
    private javax.swing.JFrame frmDelete;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JLabel lblAllocatedCapacityOnAddForm;
    private javax.swing.JLabel lblAllocatedML;
    private javax.swing.JLabel lblCapacityML;
    private javax.swing.JLabel lblCapacityOnAddForm;
    private javax.swing.JLabel lblDateRecordedOnAddForm;

```

```

private javax.swing.JLabel lblFormatOnAddForm;
private javax.swing.JLabel lblIDdeleteOnDELETEForm;
private javax.swing.JLabel lblIsActiveOnAddForm;
private javax.swing.JLabel lblProvinceOnAddForm;
private javax.swing.JLabel lblSectorOnAddForm;
private javax.swing.JLabel lblSortBy;
private javax.swing.JLabel lblSourceNameOnAddForm;
private javax.swing.JLabel lblSourceNameStartWith;
private javax.swing.JLabel lblTypeOnAddForm;
private javax.swing.JLabel lblUsedCapacityOnAddForm;
private javax.swing.JLabel lblUsedML;
private javax.swing.JTextArea txaWater;
private javax.swing.JTextField txtAllocatedCapacityOnAddForm;
private javax.swing.JTextField txtCapacityOnAddForm;
private javax.swing.JTextField txtDateRecordedOnAddForm;
private javax.swing.JTextField txtProvinceOnAddForm;
private javax.swing.JTextField txtSectorOnAddForm;
private javax.swing.JTextField txtSourceNameOnAddForm;
private javax.swing.JTextField txtSourceNameStartWith;
private javax.swing.JTextField txtTypeOnAddForm;
private javax.swing.JTextField txtUsedCapacityOnAddForm;
private javax.swing.JTextField txtIDdeleteOnDELETEForm;
// End of variables declaration//GEN-END:variables
}

```

WaterManagement Class

```

import java.util.Date;

public class WaterManagement {

    // Declaration of fields
    private int sourceID;
    private String sourceName;
    private String sourceType;
    private String province;
    private String sector;
    private int capacityML;
    private int allocatedML;
    private int usedML;
    private Date dateRecorded;
    private boolean isActive;

    // Parameterized constructor
    public WaterManagement(int id, String name, String type, String prov,
        String sect, int capacity, int allocated, int used,
        Date date, boolean active) {
        // Set the fields to the parameter values
        sourceID = id;
        sourceName = name;
        sourceType = type;
        province = prov;
        sector = sect;
        capacityML = capacity;
        allocatedML = allocated;
        usedML = used;
        dateRecorded = date;
        isActive = active;
    }

    // Accessor methods (Getters)
    public int getSourceID() {
        return sourceID;
    }

    public String getSourceName() {
        return sourceName;
    }

    public String getSourceType() {
        return sourceType;
    }
}

```



```

public String getProvince() {
    return province;
}

public String getSector() {
    return sector;
}

public int getCapacityML() {
    return capacityML;
}

public int getAllocatedML() {
    return allocatedML;
}

public int getUsedML() {
    return usedML;
}

public Date getDateRecorded() {
    return dateRecorded;
}

public boolean getIsActive() {
    return isActive;
}

// Mutator methods (Setters)
public void setSourceID(int id) {
    sourceID = id;
}

public void setSourceName(String name) {
    sourceName = name;
}

public void setSourceType(String type) {
    sourceType = type;
}

public void setProvince(String prov) {
    province = prov;
}

public void setSector(String sect) {
    sector = sect;
}

public void setCapacityML(int capacity) {
    capacityML = capacity;
}

public void setAllocatedML(int allocated) {
    allocatedML = allocated;
}

public void setUsedML(int used) {
    usedML = used;
}

public void setDateRecorded(Date date) {
    dateRecorded = date;
}

public void setIsActive(boolean active) {
    isActive = active;
}

// Return a text value of whether the water source is active (Yes/No)
private String getIsActiveText() {
    return isActive ? "Yes" : "No";
}

// Calculate the remaining water in the source

```



```

        public int calcRemainingML() {
            return capacityML - usedML;
        }

        // toString method:
        // returns a neat string representation of the WaterManagement object
        public String toString() {
            return "Source ID: " + sourceID + "\n"
                + "Source Name: " + sourceName + "\n"
                + "Source Type: " + sourceType + "\n"
                + "Province: " + province + "\n"
                + "Sector: " + sector + "\n"
                + "Capacity: " + capacityML + " ML\n"
                + "Allocated: " + allocatedML + " ML\n"
                + "Used: " + usedML + " ML\n"
                + "Remaining: " + calcRemainingML() + " ML\n"
                + "Date Recorded: " + dateRecorded + "\n"
                + "Active: " + getIsActiveText();
        }
    }
}

```

WaterManagementManger Class

```

import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.*.*;

public class WaterManagementManager {
    // Create a DB object to handle the connection to the database
    private DB watermanagmentDB = new DB();
    // ResultSet stores the data returned by the database query
    private ResultSet rs;
    // Constructor
    public WaterManagementManager() {

    }

    // Start by creating a header row for the output table
    public String getWaterData(String whereClause, String sortField) {
        String out = "Source ID" + addSpaces("Source ID", 15)
            + "Source Name" + addSpaces("Source Name", 25)
            + "Source Type" + addSpaces("Source Type", 15)
            + "Province" + addSpaces("Province", 35)
            + "Sector" + addSpaces("Sector", 15)
            + "CapacityML" + addSpaces("CapacityML", 15)
            + "AllocatedML" + addSpaces("AllocatedML", 15)
            + "UsedML" + addSpaces("UsedML", 15)
            + "Date Recorded" + addSpaces("Date Recorded", 20)
            + "IsActive" + addSpaces("IsActive", 5) + "\n\n";

        // Declare and set the value of the Query
        String query = "SELECT * FROM WaterData";

        //Check if there is a WHERE clause
        if (!whereClause.isEmpty()){

            //Add if not empty
            query += " WHERE " + whereClause;
        }

        //Check if an ORDER BY clause is specific
        if (!sortField.isEmpty()){

            //Add if not empty
            query += " ORDER BY " + sortField;
        }

        try {
            // Run the query and store the results in the ResultSet
            rs = watermanagmentDB.queryDB(query);

            // Loop through each record (row) in the ResultSet

```

```

        while (rs.next()) {
            // Extract values from each column and format them with spaces
            out += rs.getInt("SourceID") + addSpaces
            (" + rs.getInt("SourceID"), 15);
            out += rs.getString("SourceName") + addSpaces
            (rs.getString("SourceName"), 25);
            out += rs.getString("Source Type") + addSpaces
            (rs.getString("Source Type"), 15);
            out += rs.getString("Province") + addSpaces
            (rs.getString("Province"), 35);
            out += rs.getString("Sector") + addSpaces
            (rs.getString("Sector"), 15);
            out += rs.getInt("CapacityML") + addSpaces
            (" + rs.getInt("CapacityML"), 15);
            out += rs.getInt("AllocatedML") + addSpaces
            (" + rs.getInt("AllocatedML"), 15);
            out += rs.getInt("UsedML") + addSpaces
            (" + rs.getInt("UsedML"), 15);
            // Only take the first 10 characters of the date (YYYY-MM-DD)
            out += rs.getString("DateRecorded").substring(0, 10) + addSpaces
            (rs.getString("DateRecorded").substring(0, 10), 20);
            // Add IsActive field (Yes/No or True/False)
            out += rs.getString("IsActive") + addSpaces
            (rs.getString("IsActive"), 5) + "\n";
        }
    } catch (SQLException e) {
        // If something goes wrong, show an error message
        JOptionPane.showMessageDialog(null, "Could not read data");
    }
    // Return the final formatted string
    return out;
}

public void addWaterDataToDB(String inSourceName, String inSourceType,
    String inProvince, String inSector, int inCapacityML,
    int inAllocatedML, int inUsedML, String inDateRecorded,
    String inIsActive)
{
    //Insert new record
    changeDB("INSERT INTO WaterData (SourceName, [Source Type], Province, Sector, "
        + "CapacityML, AllocatedML, UsedML, DateRecorded, IsActive) VALUES" + "("
        + inSourceName + "', '" + inSourceType + "', '" + inProvince + "', '"
        + inSector + "', " + inCapacityML + ", " + inAllocatedML + ", " + inUsedML
        + ", #" + inDateRecorded + "#, " + inIsActive + ")");
}

public void deleteWaterDataFromDB (int inId)
//This method will delete Water Source with a specific ID
{ // Write the updated array to the file
    changeDB ("DELETE FROM WaterData WHERE SourceID = " + inId);
}
//Note this is a helper method, private, only used in this class
private void changeDB(String inStatement){
    // This method will write the data for all water sources to the file

    try
    {
        //Send Query to DB
        System.out.println(inStatement);
        watermanagmentDB.changeDB(inStatement);
    }
    catch (SQLException e)
    {
        //Display an error if the database could not read
        JOptionPane.showMessageDialog(null, "Could not alter database");
    }
}

/**
 * Utility method that ensures all text lines up neatly by adding spaces.
 * If a string is shorter than the width, spaces are added.
 * If longer, it gets shortened and "..." is added.

```

```
    */
    private static String addSpaces(String s, int width) {
        if (s == null) {
            s = "";
        }
        if (s.length() > width) {
            s = s.substring(0, width - 3) + "..."; // Trim and add ellipsis
        }
        // Build a string of spaces to fill the gap
        String newString = "";
        for (int i = 0; i < width - s.length(); i++) {
            newString += " ";
        }
        return newString;
    }
}
```