package m6.table\_model;

import m6.ConnectionManager;

import javax.swing.\*;

import javax.swing.table.AbstractTableModel;

import java.sql.\*;

import java.util.Vector;

public class TranscationTableModel extends AbstractTableModel {

private Vector<String> columnNames = new Vector<>();

private Vector<Object[]> data;

private JFrame ui;

private String username;

public TranscationTableModel(JFrame ui, String username) {

this.username = username;

columnNames.add("Account Number");

columnNames.add("Type");

columnNames.add("Amount");

columnNames.add("Date");

data = readFromDb();

this.ui = ui;

}

private Vector<Object[]> readFromDb() {

Connection conn = ConnectionManager.getInstance().getConnection();

Vector<Object[]> v = new Vector<>();

try {

PreparedStatement ps = conn.prepareStatement(

"SELECT transaction.accountNumber, transaction.type, transaction.amount, transaction.date " +

"FROM transaction, customer " +

"WHERE customer.username=? AND customer.accountNumber=transaction.accountNumber " +

"ORDER BY transaction.date ASC"

);

ps.setString(1, username);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

String type = rs.getString("transaction.type");

String amount = rs.getString("transaction.amount");

String accountNumber = rs.getString("transaction.accountNumber");

Date date = rs.getDate("transaction.date");

v.add(new Object[]{accountNumber, type, amount, date});

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(ui, "Error! Failed to fetch data.");

}

return v;

}

public int getColumnCount() {

return columnNames.size();

}

public int getRowCount() {

return data.size();

}

public String getColumnName(int col) {

return columnNames.get(col);

}

public Object getValueAt(int row, int col) {

return data.get(row)[col];

}

public Class getColumnClass(int c) {

return getValueAt(0, c).getClass();

}

public boolean isCellEditable(int row, int col) {

return false;

}

public void setValueAt(Object value, int row, int col) {

data.get(row)[col] = value;

}

}