

DE TOUR

Xét lược đồ CSDL quan hệ trong ứng dụng quản lý “Tour du lịch” như sau:

TOUR(MaTour, TenTour, NgayKhoiHanh, SoNgay, SoDem, Gia)

Tên từ: Mỗi tuyến du lịch cần lưu trữ mã tour (MaTour), tên tour (TenTour), ngày khởi hành (NgayKhoiHanh), bao nhiêu ngày (SoNgay), bao nhiêu đêm (SoDem), giá tour (Gia)

TINH(MaTTP, TenTTP)

Tên từ: Lưu trữ các tỉnh (thành phố) trong nước, gồm có những thông tin: mã tỉnh (hoặc thành phố) (MaTTP), tên tỉnh thành phố (TenTTP)

DIEMDL(MaDDL, TenDDL, MaTTP, DacTrung)

Tên từ: Mỗi điểm du lịch cần lưu trữ mã điểm du lịch (MaDDL), tên điểm du lịch (TenDDL) và điểm này thuộc tỉnh (thành phố) nào (MaTTP), đặc trưng (DacTrung) của điểm du lịch này là gì? (chỉ lưu 1 đặc trưng chính như: 'Tắm biển', 'Leo núi', 'Mua sắm', 'Di tích'...)

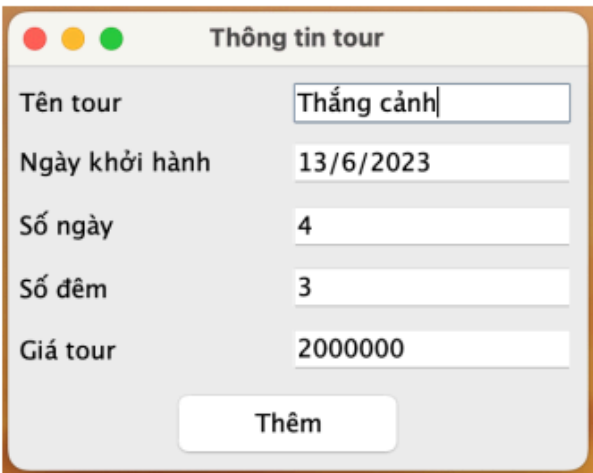
CHITIET (MaTour, MaDDL)

Tên từ: Ghi nhận mỗi Tour (MaTour) đi qua các điểm du lịch (MaDDL) nào.

Ghi chú: Các thuộc tính gạch dưới là các thuộc tính khóa chính

Sử dụng Swing để thiết kế và xử lý các chức năng sau:

1. Thêm thông tin tour du lịch (2.5 điểm)
(G3.1, G3.2, G3.3, G3.4).
Nhấn nút "**Thêm**" để thêm thông tin tour du lịch vào CSDL.
Lưu ý: MaTour được thêm tự động trong CSDL.



2. Thêm điểm du lịch (2.5 điểm) (G3.1, G3.2, G3.3, G3.4).

Thông tin về Tỉnh/TP sẽ được load từ CSDL vào combobox “Tỉnh/TP”. Nhấn nút **“Thêm”** để thêm thông tin điểm du lịch vào CSDL.

Lưu ý: MaDDL được thêm tự động trong CSDL.

3. Thêm các điểm du lịch vào tour (5 điểm) (G3.1, G3.2, G3.3, G3.4).

Nhập “Mã tour” và nhấn enter sẽ hiển thị thông tin tour gồm: tên tour, ngày khởi hành, giá tour, số ngày, số đêm (các thông tin này không cho phép chỉnh sửa).

“Tỉnh/Thành phố” được load từ CSDL vào combobox “Tỉnh/TP”. Khi chọn tỉnh/TP trong combobox sẽ hiển thị danh sách các các điểm du lịch của tỉnh/TP đó vào JTable “Điểm du lịch”.

Mỗi lần chọn điểm du lịch trong JTable “Điểm du lịch” thì điểm du lịch đó sẽ được liệt kê vào JTable “Điểm du lịch được chọn”. Nhấn nút **“Thêm”** sẽ lưu các điểm du lịch được chọn cho tour vào CSDL.

1/

```

JTextField name = new JTextField();
JTextField date = new JTextField();
JTextField day = new JTextField();
JTextField night = new JTextField();
JTextField price = new JTextField();
JButton add = new JButton("Thêm");

```

```

add.addActionListener(e -> {

```

```

        String sql = "INSERT INTO TOUR (TenTour, NgayKhoiHanh, SoNgay,
        SoDem, Gia) VALUES (?, ?, ?, ?, ?)";
        PreparedStatement pst = con.prepareStatement(sql);
        pst.setString(1, name.getText());
        pst.setString(2, date.getText());
        pst.setInt(3, Integer.parseInt(day.getText()));
        pst.setInt(4, Integer.parseInt(night.getText()));
        pst.setInt(5, Integer.parseInt(price.getText()));
        pst.executeUpdate();
    });

```

2/

```

public class TinhTP {
    public String MaTTP;
    public String TenTTP;

    public TinhTP(String MaTTP, String TenTTP) {
        this.MaTTP = MaTTP;
        this.TenTTP = TenTTP;
    }

    @Override
    public String toString() {
        return TenTTP;
    }
}

```

```

JComboBox<TinhTP> cbxTinhTP = new JComboBox<>();
JTextField txtTenDDL = new JTextField();
JTextField txtDT = new JTextField();
JButton btnAdd = new JButton("Thêm");

```

```

String sql = "SELECT MaTTP, TenTTP FROM TINHTP";
Statement st = con.createStatement();
ResultSet rs = st.executeQuery(sql);

```

```

while(rs.next()) {

```

```

        cbxTinhTP.addItem(new TinhTP(rs.getString("MaTTP"),
rs.getString("TenTTP")));
    }

    btnAdd.addActionListener(e -> {
        String sql = "INSERT INTO DIEMDL (TenDDL, MaTTP, DacTrung)
VALUES (?, ?, ?)";
        PreparedStatement pst = con.prepareStatement(sql);

        pst.setString(1, txtTenDDL.getText());
        pst.setString(2, cbxTinhTP.getSelectedItem());
        pst.setString(3, txtDT.getText());

        pst.executeUpdate();
    });

```

3/

//Load cac textfield

```

JTextField txtMaTour = new JTextField();
JTextField txtTenTour = new JTextField();
JTextField txtSoNgay = new JTextField();
JTextField txtNgayKH = new JTextField();
JTextField txtGia = new JTextField();
JTextField txtSoDem = new JTextField();

txtTenTour.setEditable(false);
txtSoNgay.setEditable(false);
txtNgayKH.setEditable(false);
txtGia.setEditable(false);
txtSoDem.setEditable(false);

txtMaTour.addActionListener(e -> {
    String sql = "SELECT * FROM TOUR WHERE MaTour = ?";
    PreparedStatement pst = con.prepareStatement(sql);

    pst.setString(1, txtMaTour.getText());
    ResultSet rs = pst.executeQuery();

```

```

        while(rs.next()) {
            txtTenTour.setText(rs.getString("TenTour"));
            txtSoNgay.setText(String.valueOf(rs.getInt("SoNgay")));
            txtNgayKH.setText(rs.getString("NgayKhoiHanh"));
            txtGia.setText(String.valueOf(rs.getInt("Gia")));
            txtSoDem.setText(String.valueOf(rs.getInt("SoDem")));
        }
    });

```

//Load ComboBox

```

JComboBox<TinhTP> cbxTinhTP = new JComboBox<>();

```

```

String sql = "SELECT MaTTP, TenTTP FROM TINHTP";
Statement st = con.createStatement();
ResultSet rs = st.executeQuery(sql);

```

```

while(rs.next()) {
    cbxTinhTP.addItem(new TinhTP(rs.getString("MaTTP"),
rs.getString("TenTTP")));
}

```

//Load 2 table

```

JTable tblDiemDL = new JTable();
JTable tblDaChon = new JTable();

```

```

String[] c1 = {"Ten diem du lich", "Dac trung"};
String[] c2 = {"Ten diem du lich"};

```

```

DefaultTableModel modelDDL = new DefaultTableModel(c1, 0);
DefaultTableModel modelDC = new DefaultTableModel(c2, 0);

```

```

tblDiemDL.setModel(modelDDL);
tblDaChon.setModel(modelDC);

```

//Load bang dau tien

```

cbxTinhTP.addActionListener(e -> {

```

```
TinhTP tinh = (TinhTP) cbxTinhTP.getSelectedItem();
String sql = "SELECT * FROM DIEMDL WHERE MaTTP = ?"
PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, tinh.MaTTP);

ResultSet rs = pst.executeQuery();
DefaultTableModel model = (DefaultTableModel) tblDiemDL.getModel();
model.setRowCount(0);

while (rs.next()) {
    model.addRow(new Object[]{
        rs.getString("TenDDL"),
        rs.getString("DacTrung")
    });
}
});
```

//Load bang thu 2

```
tblDiemDL.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        Int selectedRow = tblDiemDL.getSelectedRow();
        if (selectedRow != -1) {
            DefaultTableModel modelDL = (DefaultTableModel)
tblDiemDL.getModel();
            DefaultTableModel modelDC = (DefaultTableModel)
tblDaChon.getModel();

            Object[] rowData = new Object[]{
                modelDL.getValueAt(selectedRow, 0),
                modelDL.getValueAt(selectedRow, 1)
            };

            for (int i = 0; i < modelDC.getRowCount(); i++) {
                if (modelDaChon.getValueAt(i, 0).equals(rowData[0]))
                    return;
            }

            modelDaChon.addRow(rowData);
        }
    }
});
```

//Xu li su kien nut "THEM"

```
btnAdd.addActionListener(e -> {
    DefaultTableModel modelDC = (DefaultTableModel) tblDaChon.getModel();
    if (modelDC.getRowCount() == 0) {
        JOptionPane.showMessageDialog(null, "Vui long chon 1 DDL");
        return;
    }
    for(int i = 0; i < modelDC.getRowCount(); i++) {
        String sql = "SELECT MaDDL FROM DIEMDL WHERE TenDDL =
?";

        PreparedStatement pst = con.prepareStatement(sql);
        pst.setString(1, tblDaChon.getValueAt(i, 0).toString());
    }
});
```

```
ResultSet rs = pst.executeQuery();
if (rs.next()) {
    String MaDDL = rs.getString("MaDDL");
    String sql1 = "INSERT INTO CHITIET(MaTour, MaDDL)
VALUES (?, ?)";
    PreparedStatement pst1 = con.prepareStatement(sql1);
    pst1.setString(1, txtMaTour.getText());
    pst1.setString(2, MaDDL);

    pst1.executeUpdate();
}
});
```


DE DIEN THOAI

Xét lược đồ quan hệ trong CSDL quản lý sửa chữa điện thoại như sau:

- **DIENTHOAI(IMEI, TENDT, LOAIDT, HOTENKH, SDTKH)**

Tên từ: Lưu trữ thông tin điện thoại gồm: số IMEI của điện thoại, tên, loại điện thoại, họ tên (HOTENKH) và số điện thoại (SDTKH) của khách hàng.

- **SUACHUA(MAPSC, NGAYLAP, NOIDUNG, IMEI, TONGTIEN)**

Tên từ: Lưu trữ thông tin phiếu sửa chữa của điện thoại gồm: mã phiếu sửa chữa (MAPSC), ngày lập, nội dung sửa chữa, số IMEI của điện thoại, tổng tiền sửa chữa.

- **CONGVIEC(MACV, TENCV, DONGIA)**

Tên từ: Lưu trữ thông tin các công việc sửa chữa cho điện thoại.

- **CT_SC(MAPSC, MACV).**

Tên từ: Lưu trữ thông tin về chi tiết sửa chữa của điện thoại.

Thiết kế và xử lý các chức năng sau:

a. Thêm thông tin điện thoại cần sửa chữa.

Nhấn nút “Thêm” để thêm thông tin điện thoại cần sửa chữa vào CSDL. (2.5 điểm)

Thông tin Điện thoại

Số IMEI	1234567890
Tên điện thoại	Iphone 13 Pro
Loại điện thoại	Iphone
Họ tên khách hàng	Nguyễn C
Số điện thoại KH	09xxxxxxxx

Thêm

b. Thêm phiếu sửa chữa. Khi nhập số IMEI của điện thoại và nhấn enter thì sẽ hiển thị tên điện thoại và họ tên khách hàng sở hữu điện thoại (không cho phép chỉnh sửa). Nhấn nút “Thêm” để thêm thông tin phiếu sửa chữa điện thoại. (2.5 điểm)

Lưu ý: MAPSC được thêm tự động trong CSDL và TONGTIEN mặc định bằng 0.

Phiếu sửa chữa

Số IMEI	1234567890
Tên điện thoại	Iphone 13 Pro
Họ tên khách hàng	Nguyễn C
Ngày lập	12/6/2022
Nội dung sửa chữa	Thay màn hình Thay pin

Thêm

c. Thêm thông tin sửa chữa điện thoại.

Khi nhập số IMEI và nhấn enter thì sẽ hiển thị tên điện thoại và họ tên khách hàng sở hữu điện thoại đó. Nhập thông tin ngày lập và nhấn enter sẽ hiển thị ra nội dung sửa chữa của điện thoại với số IMEI ở trên. Các thông tin: tên điện thoại, họ tên khách hàng và nội dung sửa chữa không cho phép thay đổi.

Load danh sách công việc từ CSDL vào JTable “Danh sách công việc”; và mỗi lần chọn công việc trong danh sách này thì công việc đó sẽ được hiển thị trên JTable “Danh sách công việc sửa chữa” cùng với đơn giá của công việc đó. Nhấn nút “Thêm sửa chữa” thì sẽ lưu danh sách công việc sửa chữa này vào CSDL, đồng thời lưu tổng tiền vào phiếu sửa chữa của điện thoại đang sửa chữa. (5.0 điểm)

Lưu ý: *TONGTIEN sẽ được tính bằng tổng DONGIA của tất cả công việc trong danh sách công việc sửa chữa.*

Thông tin sửa chữa

Số IMEI: 1234567890 Tên điện thoại: Iphone 13 Pro

Họ tên KH: Nguyễn C Nội dung sửa chữa: Thay màn hình, Thay pin

Ngày lập: 12/6/2022

Mã công việc	Công việc
CV01	Thay pin Iphone
CV02	Thay pin Samsung
CV03	Thay màn hình Iphone
CV04	Sửa main Motorola

Công việc	Đơn giá
Thay pin Iphone	600000
Thay màn hình Iphone	2000000

Thêm sửa chữa

1/

```
TextField txtImei = new TextField();
TextField txtTendt = new TextField();
ComboBox cbxLoaidt = new ComboBox();
TextField txtHoten = new TextField();
TextField txtSdt = new TextField();
Button btnThem = new Button("Them");
```

```
btnThem.addActionListener(e -> {
    String sql = "INSERT INTO DIENTHOAI (IMEI, TENDT, LOAIDT,
HOTENKH, SDTKH) VALUES (?, ?, ?, ?, ?);";
```

```

        PreparedStatement pst = con.prepareStatement(sql);
        pst.setString(1, txtImei.getText());
        pst.setString(2, txtTenddt.getText());
        pst.setString(3, cbxLoaidt.getSelectedItem().toString());
        pst.setString(4, txtHoten.getText());
        pst.setString(5, txtSdt.getText());

        pst.executeUpdate();
    });

2/
JTextField txtImei = new JTextField();
JTextField txtTenddt = new JTextField();
txtTenddt.setEditable(false);
JTextField txtHoten = new JTextField();
txtHoten.setEditable(false);
JTextField txtNgaylap = new JTextField();
JTextArea txtNoidung = new JTextArea();
JButton btnThem = new JButton();

txtImei.addActionListener(e -> {
    String sql = "SELECT TENDT, HOTENKH FROM DIENTHOAI WHERE
IMEI = ?";
    PreparedStatement pst = con.prepareStatement(sql);
    pst.setString(1, txtImei.getText());

    ResultSet rs = pst.executeQuery();
    if (rs.next()){
        txtTenddt.setText(rs.getString("TENDT"));
        txtHoten.setText(rs.getString("HOTENKH"));
    }
});

btnThem.addActionListener(e -> {
    String sql = "INSERT INTO SUACHUA (NGAYLAP, NOIDUNG, IMEI)
VALUES (?, ?, ?)";
    PreparedStatement pst = con.prepareStatement(sql);

```

```
        pst.setString(1, txtNgaylap.getText());
        pst.setString(2, txtNoidung.getText());
        pst.setString(3, txtImei.getText());

        pst.executeUpdate();
    });
```

3/

```
JTextField txtImei = new JTextField();
JTextField txtTendat = new JTextField();
txtTendat.setEditable(false);
JTextField txtHoten = new JTextField();
txtHoten.setEditable(false);
JTextArea txtNoidung = new JTextArea();
txtNoidung.setEditable(false);
JTextField txtNgaylap = new JTextField();
```

```
JTable tblCV = new JTable();
JTable tblCVSC = new JTable();
```

```
String[] c1 = {"Ma cong viec", "Cong viec"};
String[] c2 = {"Cong viec", "Don gia"};
```

```
DefaultTableModel modelCV = new DefaultTableModel(c1, 0);
DefaultTableModel modelCVSC = new DefaultTableModel(c2, 0);
```

```
tblCV.setModel(modelCV);
tblCVSC.setModel(modelCVSC);
```

```
JButton btnThem = new JButton("Them sua chua");
```

//Xu li imei

```
txtImei.addActionListener(e -> {
    String sql = "SELECT TENDAT, HOTENKH FROM DIENTHOAI WHERE
IMEI = ?";
    PreparedStatement pst = con.prepareStatement(sql);
```

```

        pst.setString(1, txtImei.getText());
        ResultSet rs = pst.executeQuery();
        if (rs.next()){
            txtTendt.setText(rs.getString("TENDT"));
            txtHoten.setText(rs.getString("HOTENKH"));
        }
    });

```

//Xu li noi dung

```

txtNgaylap.addActionListener(e -> {
    String sql = "SELECT NOIDUNG FROM SUACHUA WHERE NGAYLAP =
? AND IMEI = ?";
    PreparedStatement pst = con.prepareStatement(sql);
    pst.setString(1, txtNgaylap.getText());
    pst.setString(2, txtImei.getText());

    ResultSet rs = pst.executeQuery();
    if (rs.next()){
        txtNoiDung.setText(rs.getString("NOIDUNG"));
    }
});

```

// Xu li table CV

```

String sql = "SELECT MACV, TENCV FROM CONGVIEC";
Statement st = con.createStatement();
ResultSet rs = st.executeQuery(sql);

DefaultTableModel model = (DefaultTableModel) tblCV.getModel();
model.setRowCount(0);

while (rs.next()) {
    model.addRow(new Object[]{
        rs.getString("MACV"),
        rs.getString("TENCV")
    });
}

```

//Xu li table DSCV

```
tblCV.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        int selectedRow = tblCV.getSelectedRow();
        if (selectedRow != -1){
            DefaultTableModel modelCV = (DefaultTableModel)
tblCV.getModel();
            DefaultTableModel modelDSCV = (DefaultTableModel)
tblDSCV.getModel();

            String sql = "SELECT DONGIA FROM CONGVIEC WHERE
TENCV = " + modelCV.getValueAt(selectedRow, 1) + ",";
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(sql);

            while(rs.next()){
                modelDSCV.addRow(new Object[] {
                    modelCV.getValueAt(selectedRow, 1),
                    rs.getInt("DONGIA")
                });
            }
        }
    }
});
```

//xu li nut them

```
btnThem.addActionListener(e -> {
    String maPSC = "";
    Int tongTien = 0;

    String sql = "SELECT MAPSC FROM SUACHUA WHERE NGAYLAP = ?
AND IMEI = ?";
    PreparedStatement pst = con.prepareStatement(sql);
    pst.setString(1, txtNgaylap.getText());
    pst.setString(2, txtImei.getText());
    ResultSet rs = pst.executeQuery();
    if (rs.next()){
```

```

        maPSC = rs.getString("MAPSC");
    }

    DefaultTableModel modelDSCV = (DefaultTableModel)
tblDSCV.getModel();

    for (int i = 0; i < modelDSCV.getRowCount(); i++) {
        String sql1 = "SELECT MACV FROM CONGVIEC WHERE TENCV
= ? ";
        PreparedStatement pst1 = con.prepareStatement(sql1);
        pst1.setString(1, modelDSCV.getValueAt(i, 0));
        ResultSet rs1 = pst1.executeQuery();

        if (rs1.next()) {
            String sql2 = "INSERT INTO CT_SC (MAPSC, MACV)
VALUES (?, ?)";
            PreparedStatement pst2 = con.prepareStatement(sql2);
            pst2.setString(1, maPSC);
            pst2.setString(2, rs1.getString("MACV"));
            pst2.executeUpdate();
        }
        tongTien += tongTien += Integer.parseInt(modelDSCV.getValueAt(i,
1).toString());
    }

    String sql3 = "UPDATE SUACHUA SET TONGTIEN = ? WHERE MAPSC
= ?";
    PreparedStatement pst3 = con.prepareStatement(sql3);
    pst3.setInt(1, tongTien);
    pst3.setString(2, maPSC);
    pst3.executeUpdate();
});

```