**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------🙢🕮🙠---------------

**ĐỒ ÁN MÔN HỌC XÂY DỰNG PHẦN MỀM**

**THEO MÔ HÌNH PHÂN LỚP**

**ĐỀ TÀI: QUẢN LÝ KHÓA HỌC**

**NHÓM**

**THÀNH VIÊN:**

3121410353 – Trần Khôi Nguyên

*Giảng viên:*

*Tp. Hồ chí minh, Ngày …. tháng …. năm ….*

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**MỤC LỤC**

[CHỨC NĂNG 1: QUẢN LÝ THÔNG TIN CÁC KHÓA HỌC ONLINE, ONSITE 3](#_Toc29861)

[1.1 Xử lý 1: hiển thị danh sách 3](#_Toc5845)

[1.2 Xử lý 2: hiển thị chi tiết 1 record (dòng) 3](#_Toc15741)

[1.3 Xử lý 3: thêm mới thông tin 3](#_Toc1807)

[1.4 Xử lý 4: sửa/ cập nhật thông tin 3](#_Toc556)

[1.5 Xử lý 5: tìm kiếm 4](#_Toc6152)

[1.6 Xử lý 6: xóa record 4](#_Toc21704)

[CHỨC NĂNG 2: QUẢN LÝ THÔNG TIN PHÂN CÔNG GIẢNG DẠY (COURSEINTRUCTOR) 4](#_Toc10176)

[2.1 Xử lý 1: hiển thị danh sách 4](#_Toc18084)

[2.2 Xử lý 2: hiển thị chi tiết 1 record (dòng) 5](#_Toc18494)

[2.3 Xử lý 3: thêm mới thông tin 5](#_Toc30605)

[2.4 Xử lý 4: sửa/ cập nhật thông tin 5](#_Toc8318)

[2.5 Xử lý 5: tìm kiếm 6](#_Toc18301)

[2.6 Xử lý 6: xóa record 6](#_Toc22967)

[CHỨC NĂNG 3: QUẢN LÝ KẾT QUẢ KHÓA HỌC (STUDENTGRADE) 6](#_Toc1879)

[3.1 Xử lý 1: hiển thị danh sách 6](#_Toc9147)

[3.2 Xử lý 2: hiển thị chi tiết 1 record (dòng) 6](#_Toc10292)

[3.3 Xử lý 3: thêm mới thông tin 7](#_Toc29065)

[3.4 Xử lý 4: sửa/ cập nhật thông tin 7](#_Toc9786)

[3.5 Xử lý 5: tìm kiếm 7](#_Toc22346)

[3.6 Xử lý 6: xóa record 8](#_Toc4893)

**Phân công việc làm và mức độ đóng góp:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **Phần trăm tham gia hoàn thành đồ án (tổng 100%)** | **Công việc** |
| 1 | Trần Khôi Nguyên |  |  |
| 2 | Nguyễn Thanh Phong |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

**CHỨC NĂNG 1: QUẢN LÝ THÔNG TIN CÁC KHÓA HỌC ONLINE, ONSITE**

## **1.1 Xử lý 1:** **hiển thị danh sách**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **1.2 Xử lý 2: hiển thị chi tiết 1 record (dòng)**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **1.3 Xử lý 3: thêm mới thông tin**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **1.4 Xử lý 4: sửa/ cập nhật thông tin**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **1.5 Xử lý 5: tìm kiếm**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **1.6 Xử lý 6: xóa record**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

# **CHỨC NĂNG 2: QUẢN LÝ THÔNG TIN PHÂN CÔNG GIẢNG DẠY (COURSEINTRUCTOR)**

## **2.1 Xử lý 1: hiển thị danh sách**

* **Sơ đồ tuần tự**

A diagram of a process

Description automatically generated

* **Màn hình giao diện**

A screenshot of a computer

Description automatically generated

* **Code 3 class: DAL, BLL, GUI**

**-DAL**

**A close-up of a person's face

Description automatically generated**

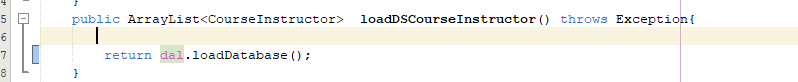
**A screenshot of a computer program

Description automatically generated**

**-BLL**

**A computer code with text

Description automatically generated with medium confidence**

****

**-GUI**

**HomePageGUI**

**A screen shot of a computer

Description automatically generated**

A computer code with text

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

## **2.2 Xử lý 2: hiển thị chi tiết 1 record (dòng)**

* **Sơ đồ tuần tự**

A diagram of a flowchart

Description automatically generated

* **Màn hình giao diện**

**A screenshot of a computer

Description automatically generated**

* **Code 3 class: DAL, BLL, GUI**

**-DAL**

**A computer screen shot of a program

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**- BLL**

**File CourseBLL**

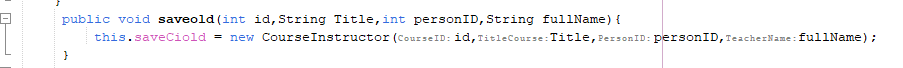
**A computer screen shot of a code

Description automatically generated**

**File LecturesBLL**

**A computer screen shot of a program

Description automatically generated**

****

**- GUI**

**HomePageGUI**

**A screen shot of a computer

Description automatically generated**

A computer code with text

Description automatically generated

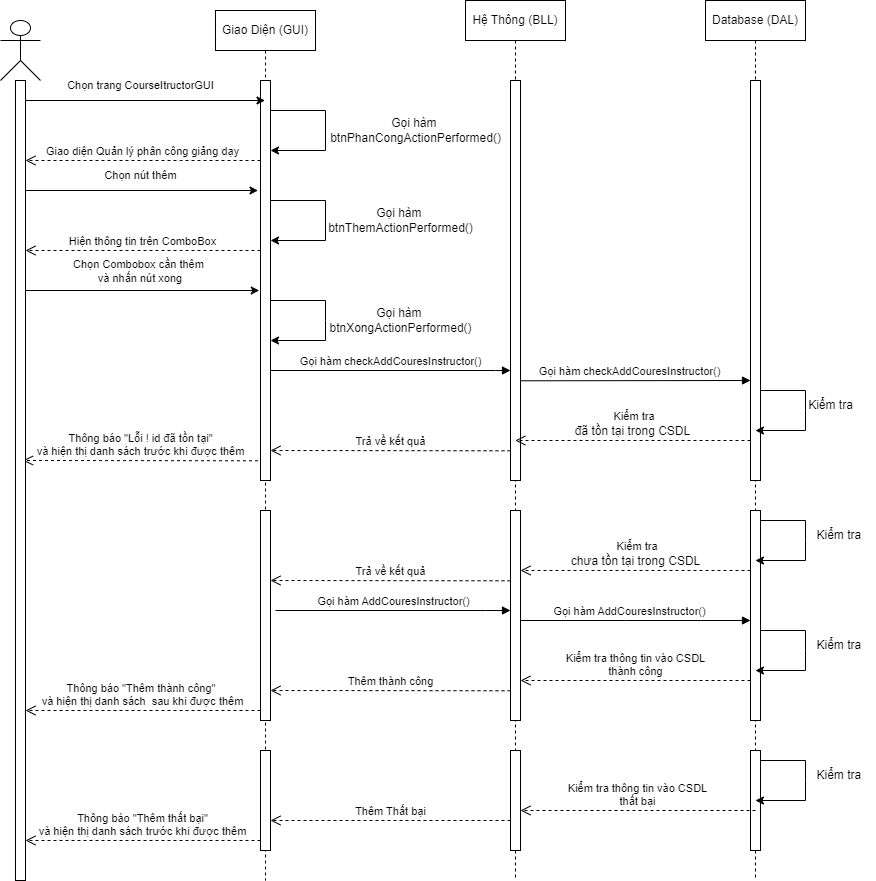
A screenshot of a computer code

Description automatically generated

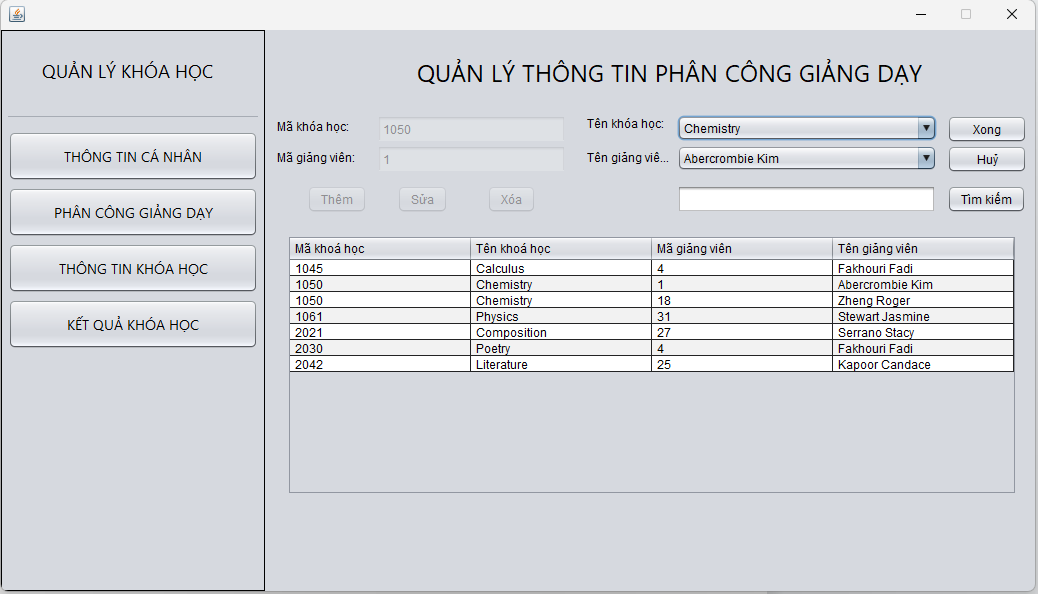
## 

## **2.3 Xử lý 3: thêm mới thông tin**

* **Sơ đồ tuần tự**



* **Màn hình giao diện**



* **Code 3 class: DAL, BLL, GUI**

**DAL: CourseIntructorDAL.java**

public boolean addCourseInstructor(int CourseID, int PersonID) throws SQLException {

        String query = "INSERT INTO CourseInstructor (CourseID, PersonID) VALUES (?, ?)";

        try (PreparedStatement p = c.prepareStatement(query)) {

            p.setInt(1, CourseID);

            p.setInt(2, PersonID);

            int rowsAffected = p.executeUpdate();

            return rowsAffected > 0;

        }

}

public int checkAddCourseInstructor(int CourseID, int PersonID) throws SQLException {

        String query = "SELECT COUNT(\*) FROM CourseInstructor WHERE CourseID = ? AND PersonID = ?";

        try (PreparedStatement p = c.prepareStatement(query)) {

            p.setInt(1, CourseID);

            p.setInt(2, PersonID);

            ResultSet rs = p.executeQuery();

            rs.next();

            int count =rs.getInt(1);

            return count ;

        }

  }

**BLL: CourseIntructorBLL.java**

public static boolean addCourseInstructor( int idcorse, int idperson) throws SQLException {

        return dal.addCourseInstructor(  idcorse, idperson);

}

public static int checkAddCourseInstructor( int idcourse, int idperson) throws SQLException {

        return dal.checkAddCourseInstructor( idcourse, idperson);

}

**GUI: CourseIntructorGUI.java**

btnThem.setText("Thêm");

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

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

btnThemActionPerformed(evt);

}

});

pprivate void btnThemActionPerformed(java.awt.event.ActionEvent evt) {

        int luaChon = JOptionPane.showConfirmDialog(null, "Có thêm không?", "Hỏi", JOptionPane.YES\_NO\_OPTION);

        if (luaChon == JOptionPane.YES\_OPTION) {

            insert = true;

            nameCourse.setEnabled(true);

            namePerson.setEnabled(true);

            btnThem.setEnabled(false);

            btnSua.setEnabled(false);

            btnXoa.setEnabled(false);

            btnXong.setVisible(true);

            btnHuy.setVisible(true);

        }

}

private void btnXongActionPerformed(java.awt.event.ActionEvent evt)

{//GEN-FIRST:event\_btnXongActionPerformed

        // TODO add your handling code here:

        if(!insert) {

            int maKhoahoc = Integer.parseInt(maKH.getText());

            int maGiangvien = Integer.parseInt(maGV.getText());

            try {

                CourseInstructor a =  cIBLL.returnOld();

                CourseInstructor b =  cIBLL.getCourseInstructorNewByID(courseBLL.findElement(maKhoahoc),personBLL.findElement(maGiangvien));

                if(!cIBLL.updateCourseInstructor(a, b)){

                    JOptionPane.showMessageDialog( null, "Update fail!!!?" );

                }else{

                  JOptionPane.showMessageDialog( null, "Update success!!!?" );

                }

                txtSearch.setText("");

                model.setRowCount(0);

                model = addArrayListToTable(cIBLL.loadDSCourseInstructor());

                tableCourseIns.setModel(model);

                btnHuyActionPerformed(evt);

            } catch (Exception ex) {

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

            }

        }

        else {

            int maKhoahoc = Integer.parseInt(maKH.getText());

            int maGiangvien = Integer.parseInt(maGV.getText());

            try {

                if (CourseInstructorBLL.checkAddCourseInstructor(maKhoahoc, maGiangvien) == 0) {

                    if(CourseInstructorBLL.addCourseInstructor(maKhoahoc, maGiangvien)) {

                        JOptionPane.showMessageDialog(null, "Thêm thành công!");

                    }

                    else {

                        JOptionPane.showMessageDialog(null, "Thêm thất bại!");

                    }

                } else {

                    JOptionPane.showMessageDialog(null, "Lỗi! id đã tồn tại");

                }

                // Cập nhật bảng hiển thị

                txtSearch.setText("");

                model.setRowCount(0);

                model = addArrayListToTable(cIBLL.loadDSCourseInstructor());

                tableCourseIns.setModel(model);

                nameCourse.setEnabled(false);

                namePerson.setEnabled(false);

                btnThem.setEnabled(!false);

                btnSua.setEnabled(!false);

                btnXoa.setEnabled(!false);

                btnXong.setVisible(!true);

                btnHuy.setVisible(!true);

            } catch (Exception e) {

                // Xử lý ngoại lệ nếu có

                e.printStackTrace();

                JOptionPane.showMessageDialog(null, "Đã xảy ra lỗi: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

            }

        }

}

## **2.4 Xử lý 4: sửa/ cập nhật thông tin**

* **Sơ đồ tuần tự**

A diagram of a diagram

Description automatically generated

* **Màn hình giao diện**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* **Code 3 class: DAL, BLL, GUI**

**-DAL**

**A computer screen shot of a code

Description automatically generated**

**-BLL**

**A white background with black text and black letters

Description automatically generated**

**A computer code with text

Description automatically generated**

**A computer screen shot of a code

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**-GUI**

**HomePageGUI**

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

A screen shot of a computer

Description automatically generated

## **2.5 Xử lý 5: tìm kiếm**

* **Sơ đồ tuần tự**

A diagram of a diagram

Description automatically generated

* **Màn hình giao diện**

**A screenshot of a computer

Description automatically generated**

* **Code 3 class: DAL, BLL, GUI**

**-DAL**

**A close-up of a person's face

Description automatically generated**

A screenshot of a computer program

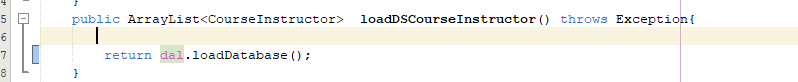
Description automatically generated

**A screenshot of a computer program

Description automatically generated**

**-BLLA computer code with black and white text

Description automatically generated**

****

**A computer code with text

Description automatically generated with medium confidence**

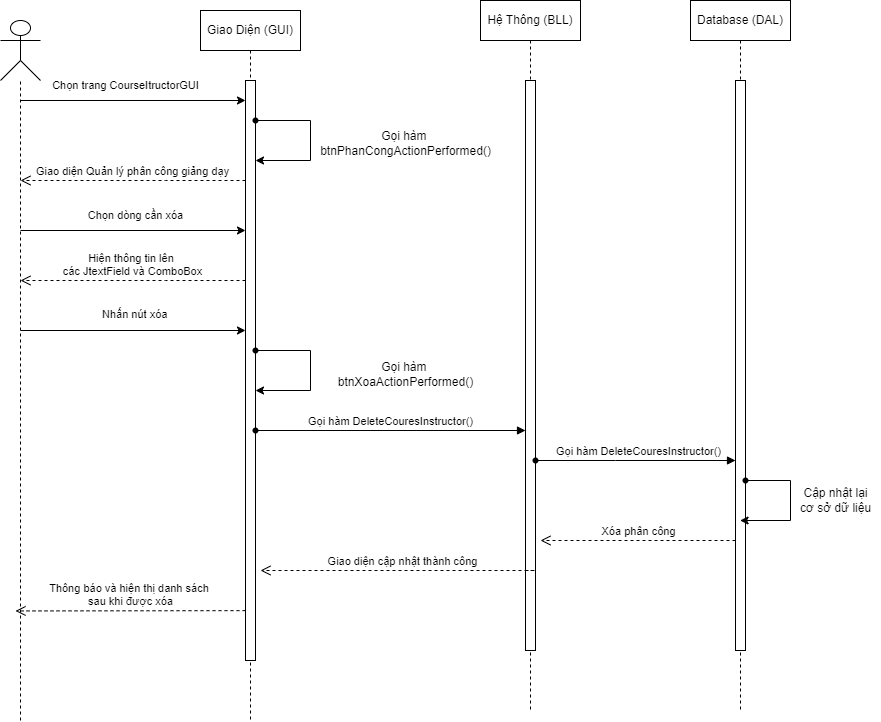
**-GUI**

A screenshot of a computer

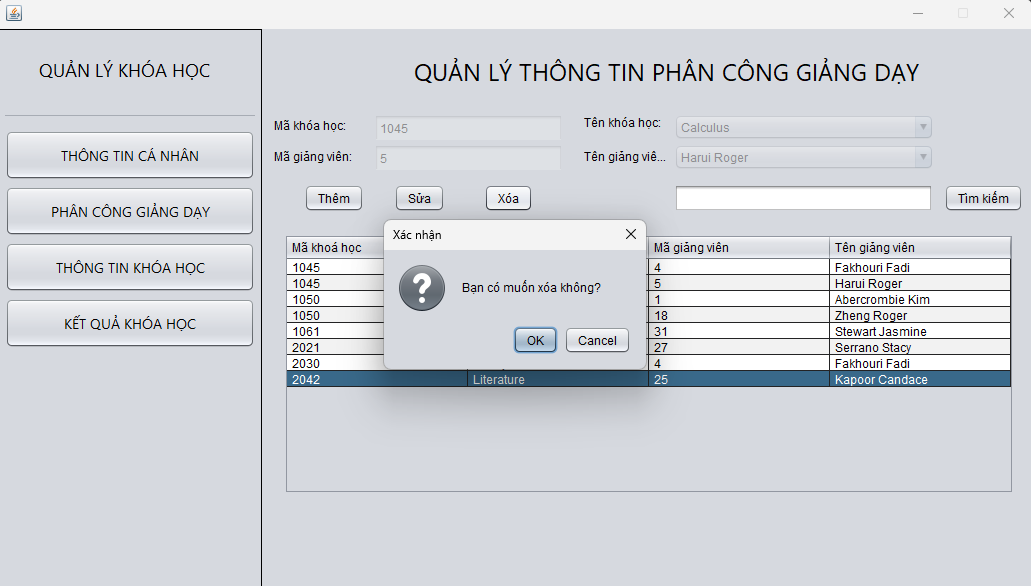
Description automatically generated

## **2.6 Xử lý 6: xóa record**

* **Sơ đồ tuần tự**



* **Màn hình giao diện**



* **Code 3 class: DAL, BLL, GUI**

**DAL: CourseIntructorDAL.java**

public int deleteCourseInstructor(int CourseID, int PersonID) throws SQLException {

        String query = "DELETE FROM CourseInstructor WHERE CourseID = ? And PersonID = ?";

        PreparedStatement p = c.prepareStatement(query);

        p.setInt(1, CourseID);

        p.setInt(2, PersonID);

        int result = p.executeUpdate();

        return result;

 }

**BLL: CourseIntructorBLL.java**

public static int deleteCourseInstructor( int idcorse, int idperson) throws SQLException {

        return dal.deleteCourseInstructor(  idcorse, idperson);

}

**GUI: CourseIntructorGUI.java**

btnXoa.setText("Xóa");

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

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

                btnXoaActionPerformed(evt);

            }

        });

        getContentPane().add(btnXoa, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 180, -1, -1));

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

        int row = (int) ((JTable) this.tableCourseIns).getSelectedRow();

        if (row < 0) {

            JOptionPane.showMessageDialog(null, "Vui lòng chọn khóa học muốn xóa");

        } else {

            int result = JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa không?", "Xác nhận", JOptionPane.OK\_CANCEL\_OPTION);

            if (result == JOptionPane.OK\_OPTION) {

                try {

                    if (CourseInstructorBLL.deleteCourseInstructor(Integer.parseInt(maKH.getText()),Integer.parseInt(maGV.getText())) != 0 )  {

                        JOptionPane.showMessageDialog(null, "Xóa thành công!");

                        model = addArrayListToTable(cIBLL.loadDSCourseInstructor());

                        tableCourseIns.setModel(model);

                    } else {

                        JOptionPane.showMessageDialog(null, "Xóa thất bại!");

                    }

                } catch (Exception e) {

                    // TODO Auto-generated catch block

                    e.printStackTrace();

                }

            }

        }

 }

# **CHỨC NĂNG 3: QUẢN LÝ KẾT QUẢ KHÓA HỌC (STUDENTGRADE)**

## **3.1 Xử lý 1: hiển thị danh sách**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **3.2 Xử lý 2: hiển thị chi tiết 1 record (dòng)**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **3.3 Xử lý 3: thêm mới thông tin**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **3.4 Xử lý 4: sửa/ cập nhật thông tin**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **3.5 Xử lý 5: tìm kiếm**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**

## **3.6 Xử lý 6: xóa record**

* **Sơ đồ tuần tự**

* **Màn hình giao diện**

* **Code 3 class: DAL, BLL, GUI**