**Hệ thống đăng ký đồ án môn học**

**ĐỀ TÀI 4 – NHÓM 12**

*Phiên bản 1.7*

6/2/2012

**Hệ quản trị CSDL TH2009/1**

GVLT: Tuấn Nguyên Hoài Đức

GVTH: Tiết Gia Hồng

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Mail | Nhóm trưởng |
| 0812266 | Trương Hoàng Liêm | [hoangliem236@gmail.com](mailto:hoangliem236@gmail.com) |  |
| 0812506 | Nguyễn Minh Thuận | [thuan178@gmail.com](mailto:thuan178@gmail.com) |  |
| 0812508 | Trần Hưng Thuận | [0812508@gmail.com](mailto:0812508@gmail.com) | X |
| 0812527 | Huỳnh Công Toàn | [7hanatos13@gmail.com](mailto:7hanatos13@gmail.com) |  |

THÔNG TIN PHIÊN BẢN

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Tác giả** | **Mô tả** |
| 09/04/2012 | 0.1 | Trương Hoàng Liêm | Liệt kế các chức năng của hệ thống |
| 09/04/2012 | 0.2 | Trần Hưng Thuận | Tạo lược đồ CSDL |
| 10/04/2012 | 0.3 | Huỳnh Công Toàn | Tạo file script CSDL |
| 15/04/2012 | 0.4 | Nguyễn Minh Thuận | Đặc tả CSDL |
| 28/04/2012 | 0.5 | Huỳnh Công Toàn | Tạo dữ liệu mẫu |
| 28/04/2012 | 0.6 | Trương Hoàng Liêm | Khảo sát tranh chấp lỗi Phantom |
| 28/04/2012 | 0.7 | Huỳnh Công Toàn | Khảo sát tranh chấp lỗi Unrepeatable Read |
| 01/05/2012 | 0.8 | Nguyễn Minh Thuận | Khảo sát tranh chấp lỗi Lost Update |
| 02/05/2012 | 1.0 | Trần Hưng Thuận | Khảo sát tranh chấp lỗi Dirty Read |
| 04/05/2012 | 1.1 | Huỳnh Công Toàn | Sửa lại khảo sát tranh chấp lỗi Unrepeatable Read |
| 06/05/2012 | 1.2 | Trương Hoàng Liêm | Sửa lại khảo sát tranh chấp lỗi Phantom |
| 06/05/2012 | 1.3 | Nguyễn Minh Thuận | Sửa lại khảo sát tranh chấp lỗi Lost Update |
| 13/05/2012 | 1.4 | Huỳnh Công Toàn | Tạo giao diện chính cho chương trình ứng dụng |
| 20/05/2012 | 1.5 | Nguyễn Minh Thuận | Xây dựng màn hình đăng nhập |
| 24/05/2012 | 1.6 | Huỳnh Công Toàn | Xây dựng màn hình cập nhật thời gian nộp đồ án |
| 29/05/2012 | 1.7 | Trương Hoàng Liêm | Xây dựng màn hình đăng ký đồ án |
|  |  |  |  |

MỤC LỤC

[**I.** **CHỨC NĂNG HỆ THỐNG** 3](#_Toc326341939)

[1) Phân hệ admin 3](#_Toc326341940)

[2) Phân hệ giáo viên 3](#_Toc326341941)

[3) Phân hệ sinh viên 3](#_Toc326341942)

[4) Các chức năng chung 3](#_Toc326341943)

[**II.** **LƯỢC ĐỒ CSDL** 3](#_Toc326341944)

[**III.** **ĐẶC TẢ CHỨC NĂNG** 4](#_Toc326341945)

[1) Đăng ký đồ án: 4](#_Toc326341946)

[2) Cập nhật thời gian nộp đồ án: 4](#_Toc326341947)

[3) Hủy đồ án: 4](#_Toc326341948)

[4) Cập nhật số lượng sinh viên nhóm: 5](#_Toc326341949)

[**IV.** **DANH SÁCH STORE PROCEDURE** 5](#_Toc326341950)

[**V.** **KHẢO SÁT TRANH CHẤP** 5](#_Toc326341951)

[1) Lost update 5](#_Toc326341952)

[2) Dirty Read 6](#_Toc326341953)

[3) Unrepeatable Read 6](#_Toc326341954)

[4) Phantom 6](#_Toc326341955)

[**VI.** **ỨNG DỤNG DEMO** 7](#_Toc326341956)

[1) Màn hình đăng nhập 7](#_Toc326341957)

[2) Màn hình chọn 4 chức năng 7](#_Toc326341958)

[3) Màn hình chức năng hủy đồ án 7](#_Toc326341959)

[4) Màn hình đăng ký đồ án 7](#_Toc326341960)

[5) Màn hình chức năng cập nhật thời hạn nộp đồ án 7](#_Toc326341961)

[6) Màn hình thay dổi số lượng sinh viên tối đa/nhóm của môn học 7](#_Toc326341962)

1. **CHỨC NĂNG HỆ THỐNG**

## Phân hệ admin

* Tạo tài khoản người dùng: tạo giáo viên và sinh viên.
* Thêm, xóa, sửa môn học (cần có thời gian bắt đầu mở và thời gian vô hiệu hóa).
* Mở/vô hiệu hóa các môn học trong khoảng thời gian nào đó.

## Phân hệ giáo viên

* Ra bài tập.
* Ra đồ án trong một môn học mà giáo viên phụ trách: chọn số lượng sinh viên tối đa cho nhóm, chọn số lượng nhóm tối đa cho đồ án.
* Xem thông tin đăng ký đồ án/ bài tập của một môn học/ bài tập cộng điểm.
* Cập nhật trạng thái đồ án/ bài tập: deadline, số lượng nhóm tối đa, số lượng sinh viên tối đa

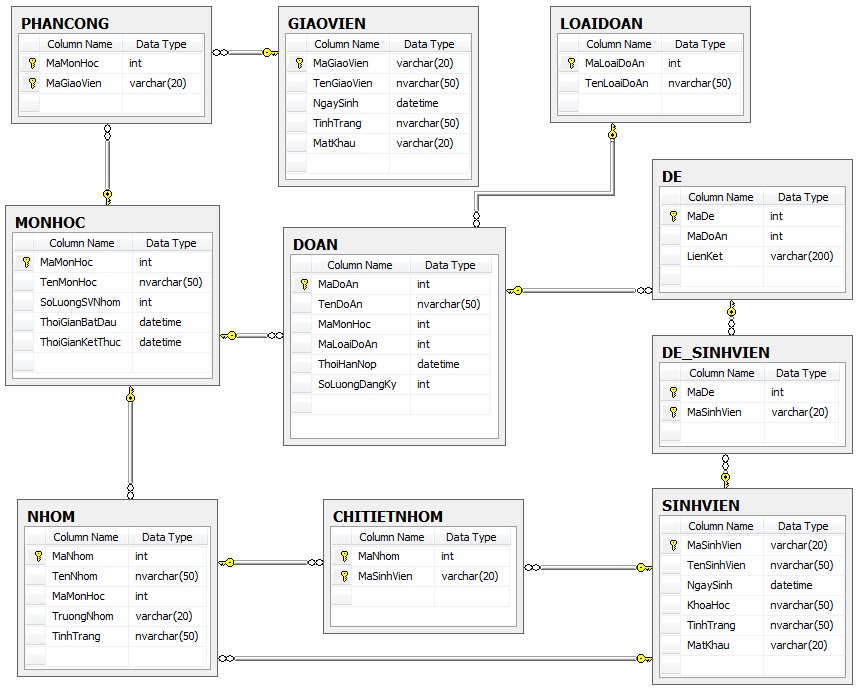
## Phân hệ sinh viên

* Đăng ký nhóm: nhóm trưởng đăng ký.
* Đăng ký đồ án: nhóm trưởng đăng ký.
* Hủy đăng ký: xem lại deadline có cho phép hay không.
* Chuyển nhóm, rút khỏi nhóm: xem lại deadline có cho phép hay không.

## Các chức năng chung

* Đăng nhập.
* Đăng xuất.

1. **LƯỢC ĐỒ CSDL**

****

1. **ĐẶC TẢ CHỨC NĂNG**

## Đăng ký đồ án:

* Sinh viên muốn đăng ký một đồ án với một đề bài cụ thể.
* Hệ thống sẽ kiểm tra thời gian nộp của đồ án đó xem có lớn hơn thời gian hiện tại và tra số nhóm đã đăng ký đồ án đó có nhỏ hơn số lương đăng ký tối đa hay không?
* Nếu thỏa hai điều kiện trên thì hệ thống sẽ thêm dữ liệu gồm mã đề và mã sinh viên vào bảng DE\_SINHVIEN.
* Sau đó hệ sẽ thống sẽ xuất thời gian nộp của đồ án và số nhóm đã đăng ký ra màn hình.

## Cập nhật thời gian nộp đồ án:

* Giáo viên có thể cập nhật thời hạn nộp của một đồ án cụ thể thuộc môn học mà giáo viên quản lý.
* Hệ thống sẽ cập nhật thời hạn nộp của đồ án này trong bảng DOAN với thời gian nộp mà giáo viên nhập.
* Hệ thống sẽ kiểm tra giáo viên nhập thời hạn nộp mới có nhỏ hơn ngày kết thúc môn học của môn học này hay không?
* Nếu không thỏa đều kiện trên thì giao tác này sẽ rollback lại trạng thái ban đầu.

## Hủy đồ án:

* Sinh viên có thể hủy đồ án mà mình đã đăng ký trước đó.
* Hệ thống sẽ kiểm tra ngày hiện tại nhỏ hơn thời hạn nộp của đồ án này.
* Nếu thỏa thì hệ thống sẽ xóa dữ liệu gồm mã đề và mã sinh viên trong bảng DE\_SINHVIEN với mã đề thuộc mã đồ án mà sinh viên muốn hủy.

## Cập nhật số lượng sinh viên nhóm:

* Một giáo viên có thể cập nhật số lượng sinh viên trong một nhóm của một môn học mà giáo viên quản lý.
* Hệ thống sẽ kiểm tra giáo viên đó phải phụ trách môn học này không?
* Nếu thỏa thì hệ thống sẽ cập nhật lại số lương sinh viên trong một nhóm của môn học này trong bảng MONHOC.

Hai giao tác cập nhật số lượng sinh viên nhóm có thể gây ra lỗi Lost Update.

Giao tác hủy đồ án và cập nhật thời hạn nộp đồ án xảy ra đồng thời có thể bị lỗi Dirly Read.

Giao tác đăng ký đồ án và cập nhật thời hạn nộp đồ án xảy ra đồng thời có thể bị lỗi Unrepeatable Read.

Hai giao tác đăng ký đồ án chạy cùng một lúc có khả năng bị lỗi Phantom.

1. **DANH SÁCH STORE PROCEDURE**

* usp\_CapNhatThoiGianNopDoAn\_Error
* usp\_DangKyDoAn\_Error
* usp\_HuyDoAn\_Error
* usp\_CapNhatSoLuongSVNhom\_Error
* usp\_CapNhatThoiGianNopDoAn\_Fix
* usp\_DangKyDoAn\_Fix
* usp\_HuyDoAn\_Fix
* usp\_CapNhatSoLuongSVNhom\_Fix
* spLogin
* usp\_SoNhomToiDa\_SoNhomDaDangKy

1. **KHẢO SÁT TRANH CHẤP**

## Lost update

* Tình huống tranh chấp xảy ra ở chức năng “Cập nhật số lượng sinh viên nhóm”.
* Tình huống tranh chấp tham chiếu đến thư mục TestCase01.
* 2 giao tác “Cập nhật số lượng sinh viên nhóm” tham chiếu đến TestCase01\_T1.sql và TestCase01\_T2.sql.
* Các hành động gây ra lỗi:
* Giáo viên có mã số “tmtriet” muốn cập nhật số lượng sinh viên nhóm của môn học “Kiến trúc phần mềm”.
* Giao tác T1 đọc số lượng sinh viên nhóm của môn “Kiến trúc hóa phần mềm”.
* Giao tác T1 tăng số lượng sinh viên nhóm lên 3 đơn vị.
* Ngay lúc T1 chưa kịp update thì giao tác T2 cũng đọc số lượng sinh viên nhóm của môn “Kiến trúc phần mềm” và thực hiện giảm số lượng sinh viên nhóm đi 1 đơn vị .
* Giao tác T1 thực hiện việc update vào CSDL.
* Sau đó giáo viên có mã số “nndkhoa” cũng muốn cập nhật số lượng sinh viên nhóm của môn “Kiến trúc phần mềm” nên thực hiện giao tác T2 đọc số lượng sinh viên nhóm của môn “Kiến trúc phần mềm” và thực hiện update vào CSDL, lúc này T2 đã không đọc kết quả của T1.
* Kết luận: số lượng sinh viên nhóm do giao tác T1 tính đã không được cập nhật.
* Cách khắc phục: sử dụng mức cô lập “repeatable read” và khóa (UPDLOCK) trên giao tác T1 và T2. Khi đó T1 trước khi đọc sẽ xin khóa và khóa được giữ đến hết giao tác này. Giao tác T2 sẽ chờ T1 trả khóa sau đó mới thực hiện việc cập nhật số lượng sinh viên nhóm. Do vậy số lượng sinh viên nhóm do giao tác T1 tính sẽ được cập nhật trước rồi mới tới T2 cập nhật.

## Dirty Read

* Tình huống tranh chấp xảy ra ở chức năng cập nhật thời hạn nộp và hủy đồ án.
* Tình huống tranh chấp tham chiếu đến thư mục TestCase02.
* Giao tác cập nhật thời hạn nộp tham chiếu đến TestCase02\_T1.sql.
* Giao tác hủy đồ án tham chiếu đến TestCase02\_T2.sql.
* Các hành động gây ra lỗi:
* Giáo viên có mã giáo viên ‘tnhduc’ muốn cập nhật thời hạn nộp của đồ án có mã đồ án ‘2’ thuộc môn học có mã ‘2’ thành ngày 7/8/2012( Deadline hiện tại là ngày 8/4/2012).
* Do đó hệ thống sẽ thực hiện giao tác cập nhật thời hạn nộp (T1).
* Khi T1 mới thực hiện xong bước update ThoiHanNop thành ngày ‘7/8/2012’ thì có 1 sinh viên có mã ‘0812001’ muốn hủy đồ án .Mà trước đó sinh viên này đã đăng ký đề có mã đề 1 thuộc đồ án có mã ‘2’ của môn học này.
* Do đó, hệ thống sẽ thực hiện giao tác hủy đồ án (T2). Do ngày hiện tại (27/5/2012) nhỏ hơn thời hạn nộp của đồ án này (7/8/2012) nên giao tác T2 thực hiện thành công.
* Nhưng sau đó, T1 thực hiện tiếp thao tác kiểm tra ThoiHanNop mới có thỏa điều kiện nhỏ hơn ngày kết thúc môn học. Do ThoiHanNop mới ‘7/8/2012’ lớn hơn ngày kết thúc môn học ‘5/7/2012’ nên T1 sẽ rollback và ThoiHanNop trở về ngày ‘8/4/2012’.
* Kết luận: giao tác T2 đọc dữ liệu rác.
* Cách khắc phục: ta sử dụng mức cô lập read committed trên 2 giao tác T1 và T2. Khi đó T2 sẽ chờ T1 trả khóa Write của ThoiHanNop(kết thúc T1)thì T2 mới thực hiện thao tác đọc ThoiHanNop.

## Unrepeatable Read

* Tình huống tranh chấp xảy ra ở chức năng “Đăng ký đồ án” và “Cập nhật thời gian nộp đồ án”.
* Tình huống tranh chấp tham chiếu đến thư mục TestCase03.
* Giao tác “Đăng ký đồ án” tham chiếu đến TestCase03\_T1.sql.
* Giao tác “Cập nhật thời gian nộp đồ án” tham chiếu đến TestCase03\_T2.sql
* Các hành động gây ra lỗi:
* Sinh viên có mã số ‘0812008’ muốn đăng ký đồ án có mã đồ án là 2, mã để là 1.
* Giao tác T1 đọc thời gian nộp của đồ án và lưu lại. Giả sử thời gian nộp lúc này là 30/07/2012, ngày kết thúc môn học ứng với đồ án trên là 01/08/2012.
* Giao tác T1 kiểm tra thời gian nộp đồ án xem có hợp lệ không để thêm dữ liệu vào bảng DE\_SINHVIEN.
* Ngay lúc này giao tác T2 thay đổi thời gian của đồ án (mã số 2) thành 29/07/2012.
* Sau đó T1 thấy dữ liệu hợp lệ và xuất ra màn hình thời hạn nộp đã lưu trước đó (bằng lệnh PRINT) là 30/07/2012.
* Giao tác T1 đọc lại thời gian nộp bằng lệnh SELECT. Thời gian nộp lúc này đã trở thành 29/07/2012.
* Kết luận: thời gian nộp do giao tác T1 đọc ở hai lần (trước và sau khi T2 chạy) có giá trị khác nhau.
* Cách khắc phục: sử dụng mức cô lập “repeatable read” trên giao tác T1. Khi đó T1 trước khi đọc sẽ xin khóa và khóa được giữ đến hết giao tác này. Giao tác T2 sẽ chờ T1 trả khóa sau đó mới thực hiện việc cập nhật thời gian nộp. Do vậy khi T1 xuất kết quả thời gian nộp ở hai lần sẽ ra giống nhau là 30/07/2012. Sau khi T1 kết thúc, T2 mới được phép cập nhật thời gian nộp thành 29/07/2012.

## Phantom

* Tình huống tranh chấp xảy ra ở chức năng “Đăng ký đồ án” .
* Tình huống tranh chấp tham chiếu đến thư mục TestCase04.
* Giao tác “Đăng ký đồ án 1” tham chiếu đến TestCase04\_T1.sql.
* Giao tác “Đăng ký đồ án 2” tham chiếu đến TestCase04\_T2.sql.
* Các hành động gây ra lỗi:
* Giả sử đồ án có mã đồ án là 9 (SoLuongDangKy = 1) lúc này có số nhóm đăng ký hiện tại là 0, tổng quát hóa lên là nếu đồ án có SoLuongDangKy = n, thì số nhóm hiện thời đăng ký đồ án là n-1.
* Sinh viên có mã số “0812007” muốn đăng ký đồ án có mã đồ án là 9(SoLuongDangKy = 1),mã để là 9.
* Giao tác T1 kiểm tra số lượng nhóm đã đăng ký của đồ án( kết các bảng rồi count), số lượng nhóm lúc này là 0.
* Giao tác T1 kiểm tra số lượng nhóm đã đăng ký của đồ án có nhỏ hơn SoLuongDangKy tương ứng với đồ án đó không, nếu nhỏ hơn thì thêm dữ liệu vào bảng DE\_SINHVIEN.
* Ngay lúc này giao tác T2cũng làm công việc tương tự là “Đăng ký đồ án” nhưng với sinh viên “0812009”. Giao tác T2 kiểm tra hợp lệ xong hết và commit hoàn thành, điều đó có nghĩa là đồ án có mã đồ án là 9 (SoLuongDangKy = 1) giờ đã có 1 nhóm đăng ký.
* Giao tác T1 kiểm tra lại số lượng nhóm đã đăng ký của đồ án, số nhóm đăng ký đồ án lúc này là 1.
* Kết luận: số nhóm đăng ký đồ án được tính toán ở hai lầntrong giao tác T1 khác nhau.
* Cách khắc phục: sử dụng mức cô lập “serializable” trên giao tác T1. Khi đó T1 trước khi đọc sẽ xin khóa và khóa được giữ đến hết giao tác này, giao tác T2 sẽ chờ T1 trả khóa này sau đó mới thực hiện việc cập nhật,thay đổi giá trị trên đơn vị dữ liệu. Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (khóa đọc) bằng sử dụng Key Range Lock, giao tác T2 sẽ phải chờ T1 trả khóa này sau đó mới thực hiện việc Insert trên đơn vị dữ liệu. Do vậy T1 tính toán số nhóm đăng ký của đồ án hai lần giống nhau.

1. **ỨNG DỤNG DEMO**

## Cấu hình chuỗi kết nối

* Mở file AbstractDAO.cs trong thư mục Source\ProjectRegistration\DAO bằng notepad.
* Điền vào chuỗi kết nối cần thiết trong phần nháy kép sau biến chuoiKetNoi0.
* Vào phương thức MoKetNoi ngay bên dưới, ở chỗ new SqlConnection(*xyz*), sửa *xyz* thành chuoiKetNoi0 (với *xyz* là chuỗi bất kỳ có sẵn).

## Màn hình đăng nhập

## Màn hình chọn 4 chức năng

## Màn hình chức năng hủy đồ án

## Màn hình đăng ký đồ án

## Màn hình chức năng cập nhật thời hạn nộp đồ án

## Màn hình thay dổi số lượng sinh viên tối đa/nhóm của môn học

**🙡🙞 HẾT 🙜🙣**