

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



# Hệ quản trị cơ sở dữ liệu Quản lý chuyển đi chơi của trẻ mầm non

GVHD: Nguyễn Hồ Duy Trí

Tên Nhóm: Nhóm 3

Tên thành viên:

- Mai Thụy Ánh Tuyết - 16521409
- Nguyễn Đức Tùng - 16521396
- Hà Thanh Đoàn - 16520289

TP HCM, Ngày 08 tháng 10 năm 2018



## Quản lý tài liệu

Ngày tạo: 08/10/2018	Thời gian lưu: 12/21/2018 10:41:14 AM
Nhóm: Nhóm 3	Lưu bởi: Mai Thụy Ánh Tuyết

## Lịch sử thay đổi

Người thực hiện	Ngày thực hiện	Nội dung	Phiên bản
Mai Thụy Ánh Tuyết	08/10/2018	Tạo báo cáo, chỉnh sửa báo cáo	1.0
Nguyễn Đức Tùng	01/12/2018	Làm báo cáo	2.0

## Lịch sử kiểm tra

Người kiểm tra	Ngày kiểm tra	Nhận xét/đánh giá	Phiên bản
{Người kiểm tra}	{dd/MM/yyyy}	{Nội dung nhận xét/đánh giá về phiên bản được kiểm tra}	{A.B.x}

# Mục lục

<b>1.</b>	<b>TỔNG QUAN DỰ ÁN</b>	<b>6</b>
1.1.	Đặt vấn đề	6
1.2.	Mục tiêu	7
1.3.	Phạm vi	7
1.4.	Công cụ sử dụng	7
1.5.	Từ ngữ viết tắt và thuật ngữ	8
1.6.	Tham khảo	8
1.7.	Các bên liên quan và nhân sự chính	8
1.8.	Vai trò và trách nhiệm thành viên dự án	9
<b>2.</b>	<b>ĐẶC TẢ YÊU CẦU</b>	<b>9</b>
2.1.	Khảo sát yêu cầu	10
2.2.	Đối tượng khảo sát	12
2.3.	Nội dung khảo sát	13
2.3.1.	Phỏng vấn	13
2.3.2.	Hiện trạng tổ chức	16
2.3.3.	Hiện trạng nghiệp vụ	17
2.3.4.	Phân tích yêu cầu	17
2.3.5.	Các ràng buộc của hệ thống	19
2.4.	Lựa chọn phương án	19
<b>3.</b>	<b>THIẾT KẾ - ĐẶC TẢ HỆ THỐNG</b>	<b>21</b>
3.1.	Sơ đồ Usecase	21
3.2.	Danh sách các tác nhân của hệ thống	26
3.3.	Danh sách Usecase	26
<b>4.</b>	<b>THIẾT KẾ CƠ SỞ DỮ LIỆU</b>	<b>27</b>

4.1.	Sơ đồ lớp-----	27
4.2.	Danh sách các thành phần trong sơ đồ lớp -----	28
4.3.	Mô tả chi tiết các lớp đối tượng-----	29
4.3.1.	Lớp đối tượng HocSinh-----	29
4.3.2.	Lớp đối tượng NamHoc-----	29
4.3.3.	Lớp đối tượng LopHoc-----	29
4.3.4.	Lớp đối tượng CTLop-----	30
4.3.5.	Lớp đối tượng GiaoVien-----	30
4.3.6.	Lớp đối tượng CongTyDuLich-----	31
4.3.7.	Lớp đối tượng DiaDiem-----	31
4.3.8.	Lớp đối tượng ChuyenDi-----	31
4.3.9.	Lớp đối tượng ChiTietChuyenDi-----	32
4.3.10.	Lớp đối tượng ChiPhi-----	32
4.3.11.	Lớp đối tượng HopDong-----	33
4.3.12.	Lớp đối tượng ThongTinThanhToan-----	33
4.3.13.	Lớp đối tượng HocSinhThamGia-----	34
4.3.14.	Lớp đối tượng GiaoVienThamGia-----	34
4.3.15.	Lớp đối tượng [User]-----	35
4.3.16.	Lớp đối tượng CT_GV_Lop-----	35
5.	XÂY DỰNG VÀ XỬ LÝ THAO TÁC-----	36
5.1.	Trigger-----	36
5.1.1.	Danh sách các trigger trong phần mềm-----	36
5.1.2.	Mô tả các trigger-----	39
5.2.	Store procedure-----	50
5.2.1.	Danh sách các Store procedure-----	50
5.2.2.	Mô tả các Store procedure-----	52
6.	XỬ LÝ TRUY XUẤT ĐỒNG THỜI-----	59
6.1.	Các mức cô lập trong SQL Server-----	59
6.1.1.	Read Uncommitted-----	59
6.1.2.	Read Committed-----	59
6.1.3.	Repeated Read-----	60
6.1.4.	Serializable-----	60
6.2.	Các vấn đề xảy ra khi truy vấn đồng thời-----	61

6.2.1.	Lost update	61
6.2.2.	Dirty read	71
6.2.3.	Unrepeatable read	77
6.2.4.	Phantom	82
6.3.	Deadlock	87
7.	THIẾT KẾ GIAO DIỆN	91
8.	KIỂM TRA PHẦN MỀM	99
9.	TỔNG KẾT – ĐÁNH GIÁ	104

# Lời cảm ơn

Đầu tiên nhóm xin gửi lời cảm ơn đến toàn thể quý Thầy/Cô trường Đại học Công nghệ Thông tin. Đặc biệt là quý Thầy cô khoa Hệ thống thông tin đã giúp cho nhóm có thêm các kiến thức nền tảng hoàn thành đề tài này.

Đặc biệt cảm ơn đến Thầy Nguyễn Hồ Duy Trí và cô Thái Bảo Trân đã góp ý giúp nhóm hoàn thành đồ án một cách tốt đẹp.

Trong thời gian thực hiện đề tài nhóm đã vận dụng các kiến thức được hướng dẫn cùng với các nghiên cứu kiến thức mới. Tuy nhiên trong thời gian làm đồ án ắt hẳn có những sai sót. Chính vì vậy chúng em mong muốn nhận được sự đóng góp từ phía Thầy Cô nhằm hoàn thiện các kiến thức đã học và biết áp dụng chúng trong tương lai.

## 1. Tổng quan dự án

### 1.1. Đặt vấn đề

Trong bối cảnh hiện nay, trường mầm non là một phần quan trọng của đất nước. Là nơi chăm sóc các em nhỏ trong suốt thời gian từ 2-5 tuổi. Ngoài những chương trình giảng dạy trên lớp. Trẻ mầm non còn cần phải được tiếp xúc thực tế với thế giới bên ngoài. Như chúng ta thấy hiện nay, việc trẻ em sử dụng smartphone của cha mẹ không phải là chuyện hiếm gặp. Điều này khiến các em trở nên thụ động, ham thích thế giới ảo, các bệnh tật về mắt, chậm phát triển ngôn ngữ vì không giao tiếp với người khác, không dám tiếp xúc với người ngoài. Vì thế việc cho trẻ tiếp xúc thế giới bên ngoài là điều vô cùng cấp bách. Chính vì vậy, các cô giáo trường mẫu giáo Thiên Thần đã cùng nhau lên kế hoạch cho trẻ đi khám phá thế giới thực. Tuy nhiên việc quản lý còn gặp nhiều khó khăn. Cô hiệu trưởng phải khó khăn trong việc phân chia giáo viên vào các lớp, khó khăn trong việc quản lý số lượng trẻ.....

Trường mẫu giáo Thiên Thần hiện nay vẫn chưa có một phần mềm để quản lý chuyến đi chơi của trẻ. Để giải quyết vấn đề này cần xây dựng nên một phần mềm quản lý chuyến đi chơi của trẻ. Hệ thống này là hệ thống có thể lưu trữ, truy vấn thông tin một cách tập trung, tiện lợi, dễ dàng tương tác với người sử dụng. Bên cạnh đó cũng cần chuẩn hóa quy trình các công việc tại trường để việc ứng dụng công nghệ thông tin trở nên hiệu quả.

## **1.2. Mục tiêu**

Xây dựng một phần mềm quản lý chuyến đi chơi của trẻ mầm non trường mẫu giáo thiên thần nhằm đáp ứng nhu cầu của các giáo viên trường trong việc quản lý số lượng trẻ tham gia chuyến tham quan, dễ dàng tính toán, quản lý chi phí, lấy thông tin các công ty du lịch. Dễ dàng lên kế hoạch cho chuyến đi đồng thời việc báo cáo thống kê hàng năm trở nên dễ dàng hơn.

## **1.3. Phạm vi**

Do thời gian hạn chế, cùng kiến thức chưa sâu rộng. Phần mềm quản lý chuyến đi chơi của trẻ mầm non thực hiện một số chức năng như sau:

- Quản lý thông tin chuyến đi
- Quản lý thông tin hồ sơ học sinh
- Quản lý thông tin giáo viên
- Quản lý công ty du lịch
- Quản lý địa điểm
- Quản lý chương trình đi
- Quản lý chi phí
- Quản lý hợp đồng với công ty du lịch
- Thống kê chung theo từng năm
- Quản lý liên hệ với phụ huynh, gửi email tự động.....
- Tạo phần mềm thông minh, gợi ý cho người sử dụng

## **1.4. Công cụ sử dụng**

Trong quá trình thực hiện đồ án, nhóm đã sử dụng các phần mềm sau

- Microsoft SQL Server 2014: Nơi lưu cơ sở dữ liệu và quản trị cơ sở dữ liệu
- NetBean IDE 8.2 : IDE Lập trình java cho phần mềm
- Visio 2013 : Thực hiện vẽ các sơ đồ, các mô hình tổ chức.



### 1.5. Từ ngữ viết tắt và thuật ngữ

Từ viết tắt	Diễn giải
CSDL	Cơ sở dữ liệu
MN	Mầm non

### 1.6. Tham khảo

STT	Tên tài liệu/ Nguồn tham khảo	Mô tả chi tiết
1	<a href="https://www.facebook.com/than.truongmgthien/about?lst=100007336510185%3A100026736321026%3A1544628145">https://www.facebook.com/than.truongmgthien/about?lst=100007336510185%3A100026736321026%3A1544628145</a>	Facebook trường MN Thiên Thần
2	<a href="https://drive.google.com/file/d/0BxS8gcT3zfwbahY2WXdlG9ET2M/view">https://drive.google.com/file/d/0BxS8gcT3zfwbahY2WXdlG9ET2M/view</a>	Đồ án mẫu
3	Slide bài giảng UML	

### 1.7. Các bên liên quan và nhân sự chính

STT	Họ Tên	Bộ phận	Vai trò & Liên lạc trách nhiệm
1	Mai Thị Thu Bích	Ban Giám Hiệu	Hiệu trưởng Email:ntcglm@gmail.com SDT: 0973654893
2	Nguyễn Thị Trâm Anh		Hiệu phó chuyên môn Email: <a href="mailto:tramanh@gmail.com">tramanh@gmail.com</a> SDT: 0352032844

## 1.8. Vai trò và trách nhiệm thành viên dự án

STT	Họ Tên	Vai trò	Trách nhiệm chính
1	Mai Thụy Ánh Tuyết	Nhóm trưởng	Viết báo cáo
2	Hà Thanh Đoàn	Thành viên	Thiết kế dữ liệu, sơ đồ
3	Nguyễn Đức Tùng	Thành viên	Thiết kế dữ liệu, sơ đồ, lập trình

## 2. Đặc tả Yêu cầu

Chương 1 đã giới thiệu tổng quan về dự án, phạm vi, mục tiêu, nhân sự và các bên liên quan. Từ đó giúp độc giả biết sơ qua về dự án và định hướng của dự án vào thực tiễn. Sau đây chúng ta sẽ chuyển sang một phần mới đó là đặc tả yêu cầu dự án.

Qua khảo sát tại trường mẫu giáo Thiên Thần nhóm đã ghi nhận lại các yêu cầu từ phía người sử dụng như sau

STT	Nhiệm vụ	Ghi chú
1	Quản lý học sinh	
2	Quản lý giáo viên	
3	Quản lý công ty du lịch	
4	Quản lý địa điểm	
5	Quản lý thông tin chuyến đi	
6	Quản lý chương trình chi tiết	
7	Quản lý hợp đồng	
8	Quản lý chi tiết hợp đồng	
8	Quản lý chi phí	
9	Báo cáo thống kê hàng năm	

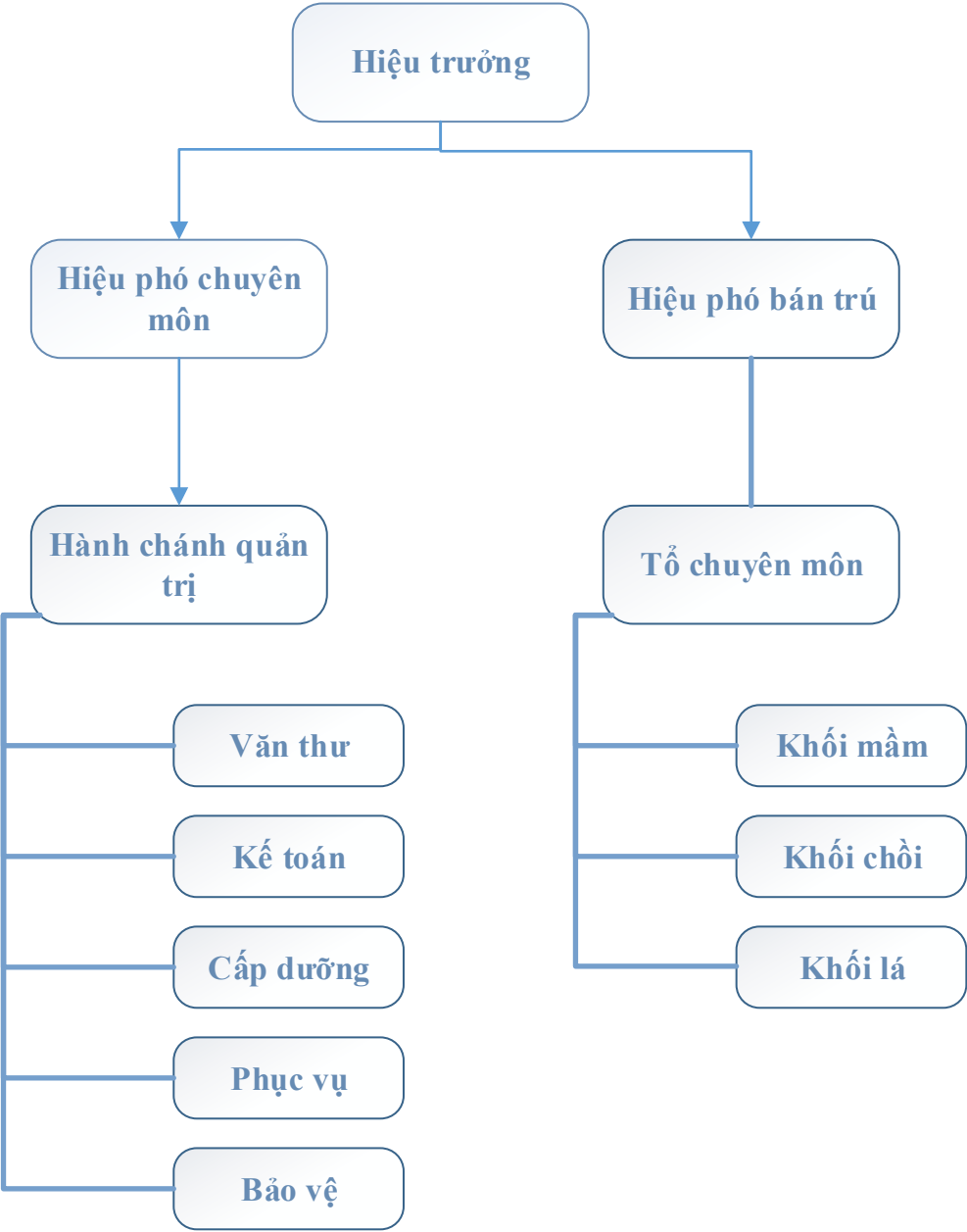
## 2.1. Khảo sát yêu cầu

STT	Chủ đề	Yêu cầu	Ngày bắt đầu	Ngày kết thúc
1	Quản lý học sinh	Nắm rõ quy trình nhập liệu hồ sơ học sinh trong nhà trường.	08/10/2018	08/10/2018
2	Quản lý giáo viên	Nắm rõ quy trình nhập liệu hồ sơ giáo viên trong nhà trường	08/10/2018	08/10/2018
3	Quản lý công ty du lịch	Hiểu các thông tin về công ty: trụ sở, liên hệ, các dịch vụ cung cấp	08/10/2018	08/10/2018
4	Quản lý địa điểm	Hiểu tất cả các thông tin liên hệ, các loại địa điểm tham quan(văn hóa, du lịch...).	08/10/2018	08/10/2018
5	Quản lý thông tin chuyến đi	Hiểu quy trình của một chuyến đi, đây là phần chủ chốt trong phần mềm. Thông tin chuyến đi gồm: công ty du lịch, địa điểm, giờ	10/10/2018	10/10/2018

		khởi hành, kết thúc		
6	Quản lý chương trình chi tiết	Hiểu các khung giờ của chương trình. Một chương trình chi tiết là một danh sách các khung giờ, công việc cần làm trong khung giờ đó	10/10/2018	10/10/2018
7	Quản lý hợp đồng	Lưu lại thông tin hợp đồng giữa hai công ty du lịch và nhà trường. phục vụ cho nhu cầu tra cứu và liên hệ với công ty	10/10/2018	10/10/2018
8	Quản lý chi phí	Hiểu các khoản tiền cần chi cho một chuyến tham quan.	10/10/2018	10/10/2018
9	Báo cáo thống kê hàng năm	Hiểu nghiệp vụ thống kê lại số lượng trẻ, chi phí theo từng năm	10/10/2018	10/10/2018

2.2.      **Đối tượng khảo sát**

Dưới đây là sơ đồ tổ chức trường mẫu giáo Thiên Thần sau khi khảo sát



## 2.3. Nội dung khảo sát

### 2.3.1. Phỏng vấn

<b>Bảng kế hoạch phỏng vấn</b>		
<b>Hệ thống: Quản lý chuyển đi chơi của trẻ mầm non trường mẫu giáo Thiên Thần</b>		
Người được phỏng vấn	Cô Mai Thị Thu Bích	Phân tích viên: Mai Thụy Ánh Tuyết
Vị trí	Phòng hiệu trưởng trường mẫu giáo thiên thần	Thời gian bắt đầu: 08/10/2018
Mục tiêu: Thu nhận yêu cầu, hiểu được quy trình của hệ thống quản lý chuyển đi chơi của trẻ mầm non		
Chi tiết buổi phỏng vấn <ul style="list-style-type: none"><li>- Giới thiệu</li><li>- Tổng quan yêu cầu của Cô hiệu trưởng</li><li>- Tổng quan về buổi phỏng vấn</li></ul>		Thời gian phỏng vấn <ul style="list-style-type: none"><li>- 5 phút</li><li>- 20 phút</li><li>- 2 giờ</li></ul>

Người được phỏng vấn	Mai Thị Thu Bích	Ngày: 08/10/2018
Câu hỏi		Ghi nhận
	Việc tổ chức cho các bé đi tham quan diễn ra vào thời gian nào	Khoảng 1 tháng/ lần
	Nhà trường đã có phần mềm để quản lý các bé chưa? Nếu có thì sử dụng những phần mềm gì?	Hiện tại nhà trường vẫn chưa có phần mềm để quản lý các bé. Mọi việc quản lý dùng Excel, Access. Hiện đang có rất nhiều khó khăn và còn nhiều bất cập
	Về việc đi tham quan của trẻ nhà trường có sử dụng phần mềm quản lý nào không?	Hiện tại nhà trường vẫn chưa có phần mềm nào để quản lý các bé, mọi kế hoạch tổ chức phân chia đều làm tay hoặc trên word rồi gửi cho giáo viên của từng lớp
	Vậy có những khó khăn gì trong quá trình quản lý trẻ?	Phần quản lý trẻ gặp nhiều khó khăn, vì bản thân tôi không biết được có bao nhiêu em học sinh tham gia ở thời điểm

		hiện tại. Mọi việc đều nhờ giáo viên điểm danh xong mới báo lại cho hiệu trưởng. Việc chờ đợi có nhiều bất cập.
	Vậy cô mong muốn có một phần mềm quản lý chuyến đi của trẻ như thế nào?	Tôi muốn có một phần mềm có thể kiểm soát được số lượng trẻ trong một năm đi tham quan được bao nhiêu em? Mỗi lần sẽ hết chi phí bao nhiêu? Các địa điểm các em đã đi qua. Thông tin các công ty du lịch. Thống kê chi phí sau một năm
	Vậy một chuyến đi mình sẽ chuẩn bị những gì ạ?	Đầu tiên mình sẽ lên kế hoạch ngày nào đi, địa điểm nào, công ty du lịch nào phụ trách. Thường sẽ đi chơi trong 1 ngày. Một chuyến đi thì thường đi các địa điểm an toàn như: sở thú, siêu thị, đầm sen, suối tiên, nơi làm bánh..... Để các bé được trải nghiệm thế giới thực.
	Vậy quy trình tổng quát một chuyến đi như thế nào?	Quy trình: <ul style="list-style-type: none"> <li>- Lên kế hoạch</li> <li>- Thông báo phụ huynh( Thông tin chuyến đi, số tiền đóng)</li> <li>- Chốt SL trẻ</li> <li>- Phân giáo viên quản lý trẻ</li> <li>- Liên hệ công ty du lịch( người ta tự sắp xe cho mình)</li> <li>- Công ty sẽ gửi về địa điểm, thực đơn, chương trình đi ( tham quan những điểm nào)</li> <li>- Xác định thời gian bắt đầu, thời gian kết thúc.</li> </ul>
	Một kế hoạch chuyến đi gồm những gì?	Một kế hoạch bao gồm: Mục đích , yêu cầu chuyến đi, timeline dự kiến( bắt đầu, kết thúc, giờ nào làm việc nào...)

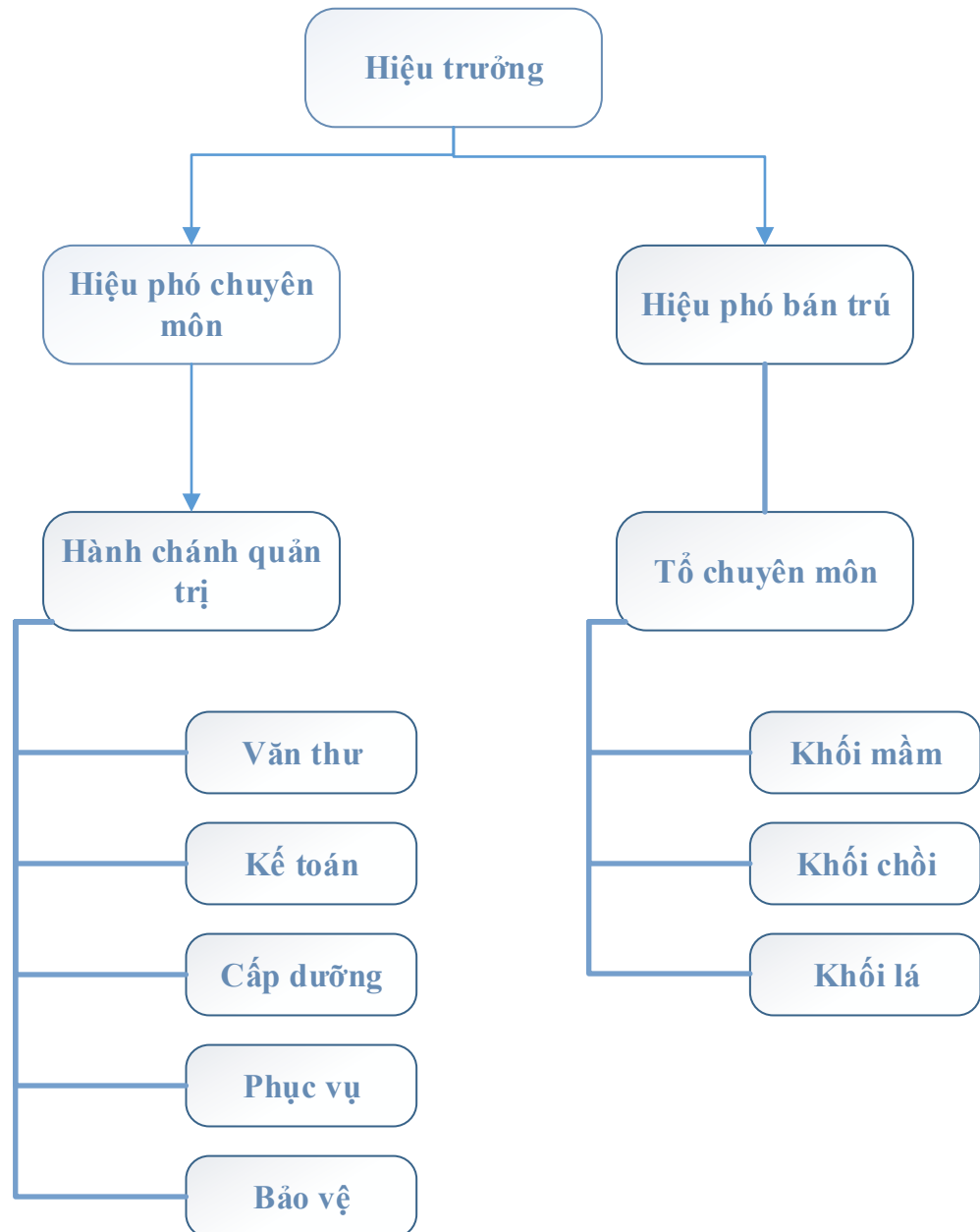
	Vậy làm thế nào để mình liên lạc với nhà xe?	Trường có một danh sách các công ty du lịch. Mỗi tháng sẽ liên hệ với công ty du lịch đó.
	Về vấn đề thực đơn trong chuyến đi đó như thế nào?	Mỗi chuyến đi, công ty du lịch sẽ gửi lại cho nhà trường, sau đó trường sẽ gửi email về số lượng trẻ, số giáo viên.
	Còn giáo viên phân chia để quản lý các lớp như thế nào?	Sau khi lên kế hoạch, hiệu trưởng cũng phân chia các giáo viên. Sau đó mới thông báo cho các giáo viên phụ trách các lớp.
	Vậy về phần quản lý chi phí, mình sẽ quản lý những gì	Về chi phí có vé cổng, tiền xe, ăn trưa, ăn xế, trò chơi, phí hướng dẫn viên, nước uống.... cho trẻ.
	Mỗi năm mình sẽ thống kê lại như thế nào	Thật khó để thống kê, mỗi năm chỉ thống kê lại được chi phí, còn số trẻ, số giáo viên thì không thể thống kê hết được
	Vậy về việc hợp đồng với các công ty mình sẽ lưu tại đâu	Hợp đồng ký theo từng năm và lưu lại trên văn phòng của nhà trường



### 2.3.2. Hiện trạng tổ chức

#### 2.3.2.1. Đối nội

Dưới đây là cơ cấu tổ chức trong trường mẫu giáo Thiên Thần



#### 2.3.2.2. Đối ngoại

- Bộ Giáo dục và Đào tạo.
- Dịch vụ kết nối internet.
- Hệ thống điện, nước.
- Liên kết với các trường khác.

### **2.3.3. Hiện trạng nghiệp vụ**

#### **2.3.3.1. Quy trình quản lý nhân sự**

- Quản lý nhân sự gồm có việc quản lý học sinh và quản lý giáo viên, phục vụ cho việc phân chia, tra cứu thông tin học sinh hoặc thông tin giáo viên.
- Mỗi năm phòng hiệu trưởng sẽ chia lại danh sách các lớp, các giáo viên

#### **2.3.3.2. Quản lý chuyến đi**

- Lên kế hoạch chuyến đi
- Thông báo đến quý phụ huynh về thông tin chuyến đi, số tiền.
- Xác định lại số lượng trẻ tham dự
- Phân chia giáo viên quản lý trẻ
- Liên hệ công ty du lịch về thời gian tổ chức, chương trình dự kiến.
- Công ty sẽ gửi về địa điểm, thực đơn, chương trình đi ( tham quan những điểm nào)
- Xác định thời gian bắt đầu, thời gian kết thúc.

### **2.3.4. Phân tích yêu cầu**

#### **2.3.4.1. Yêu cầu chức năng**

##### **2.3.4.1.1. Yêu cầu lưu trữ**

Tính năng lưu trữ là tính năng chủ chốt của phần mềm. Với những thông tin hỗ trợ chủ yếu cho việc thực hiện các chức năng của phần mềm quản lý chuyến đi chơi của trẻ mầm non. Để người quản trị có thể thực hiện dễ dàng hơn, hệ thống đã lưu lại một số thông tin sau:

##### **❖ Quản lý học sinh**

- Là việc lưu lại thông tin học sinh ở các lớp trong mỗi năm học. Trong đó có các thông tin như mã số học sinh, họ tên học sinh, ngày sinh, địa chỉ, tên cha mẹ, người giám hộ (nếu có), SĐT phụ huynh
- Quy định: độ tuổi (3-5 tuổi), họ tên....
- Nơi sử dụng: Phòng hiệu trưởng.

##### **❖ Quản lý giáo viên**

- Thông tin đầu vào: Mã giáo viên, tên giáo viên ngày sinh, địa chỉ, số điện thoại chứng minh nhân dân, mã lớp sẽ quản lý.
- Nơi sử dụng: phòng hiệu trưởng

##### **❖ Quản lý thông tin chuyến đi**

- Nghiệp vụ quản lý chuyến đi gồm có các thông tin : mã chuyến đi, thời điểm khởi hành và kết thúc, địa điểm diễn ra, công ty du lịch phụ trách.

- Nơi sử dụng: phòng hiệu trưởng.
- Mỗi chuyến đi sẽ được lên kế hoạch 1 tháng/ 1 lần.
  - ❖ Quản lý chương trình chi tiết
- Nơi quản lý các khung giờ của một chương trình: thời gian nào diễn ra công việc nào, ghi chú(nếu có).
  - ❖ Quản lý địa điểm
- Quản lý các địa điểm của một chương trình. Địa điểm gồm có các thông tin: Mã địa điểm, tên địa điểm, địa chỉ.
  - ❖ Quản lý hợp đồng
- Lưu lại thông tin hợp đồng sau khi đã ký với phía công ty du lịch. Hợp đồng gồm
  - ❖ Quản lý chi phí
- Thông tin lưu lại trong hệ thống là các khoản chi tiêu trong một chuyến đi. Thông tin quản lý gồm địa điểm, chuyến đi, các khoản: vé vào cổng, tiền xe, tiền ăn trưa, ăn xế, nước uống, phí hướng dẫn viên du lịch và các khoản khác cùng ghi chú cho các khoản ấy.

#### **2.3.4.1.2. Yêu cầu chức năng**

Ngoài việc lưu trữ thông tin, phần mềm còn hỗ trợ người sử dụng các chức năng sau:

- Chức năng tìm kiếm: giúp người sử dụng dễ dàng tìm kiếm các thông tin về: thông tin học sinh, thông tin giáo viên, thông tin chuyến đi, thông tin địa điểm.....
- Chức năng thống kê báo cáo: Giúp cô hiệu trưởng dễ dàng thống kê lại số lượng trẻ tham dự, các khoản chi tiêu, các địa điểm..... Mà không cần mất thời gian tìm kiếm qua giấy tờ tại văn phòng.
- Các chức năng tự động: giúp người dùng sử dụng phần mềm thuận tiện và dễ dàng hơn. Một số chức năng như tự động thêm mã chương trình, mã học sinh, mã giáo viên, tính toán thu chi....

#### **2.3.4.2. Yêu cầu phi chức năng**

- Yêu cầu về giao diện: phù hợp với chuyên môn ngành mầm non, thuận tiện cho việc quản lý chuyên nghiệp.
- Yêu cầu chất lượng
  - Tính tiến hóa: dễ dàng cho việc nâng cấp phần mềm, bảo trì sửa chữa phần mềm
  - Tính tiện dụng: Dễ sử dụng, ít thao tác mà hiệu quả. Có hướng dẫn sử dụng rõ ràng, minh bạch
  - Tính hiệu quả : Phần mềm thể hiện được thông tin về các chuyến đi. Hoạt động ổn định tránh lỗi không đáng tiếc xảy ra.

### 2.3.5. Các ràng buộc của hệ thống

<b>R1</b>	Giao diện rõ ràng, dễ sử dụng
<b>R2</b>	Dễ dàng truy cập đến tất cả dữ liệu cần thiết
<b>R3</b>	Thời gian phản hồi của hệ thống nhanh chóng
<b>R4</b>	Tính ổn định khi chạy đồng thời với các ứng dụng khác
<b>R5</b>	Chi phí phát triển phần mềm không vượt quá 30 triệu
<b>R6</b>	Chi phí phần cứng không được vượt quá 20 triệu
<b>R7</b>	Hệ thống phải hoạt động sau 4 tháng kể từ ngày bắt đầu
<b>R8</b>	Giới hạn chức năng của hệ thống theo từng phân quyền.

## 2.4. Lựa chọn phương án

- Các phương án đề xuất

Tiêu chuẩn	Phương án A	Phương án B	Phương án C
<b>Yêu cầu:</b>			
Y1 -> Y6	Có	Có	Có
Y7	Không	Có	Có
Y8	Không	Có	Có
Y9	Có	Có	Có
Y10	Không	Không	Có
<b>Ràng buộc:</b>			
R1	Đáp ứng	Đáp ứng	Đáp ứng
R2	Đáp ứng	Đáp ứng	Đáp ứng
R3	Đáp ứng	Đáp ứng	Đáp ứng
R4	Đáp ứng	Đáp ứng	Đáp ứng
R5	20	30	40
R6	15	22	35
R7	3	4	5.5
R8	Đáp ứng	Đáp ứng	Đáp ứng

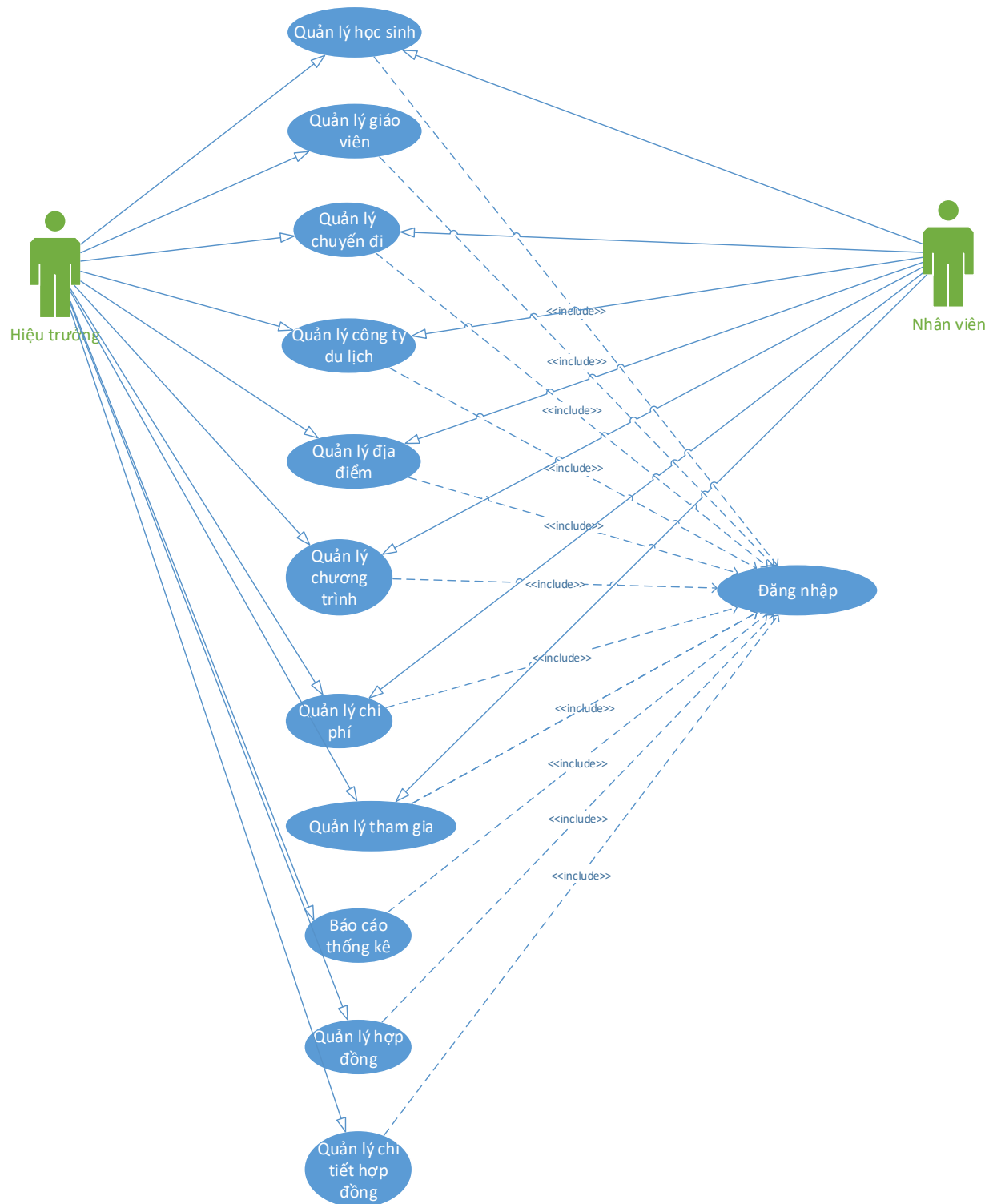
- Lựa chọn phương án

Điều kiện	Trọng số	Phương án A		Phương án B		Phương án C	
		Tỉ lệ	Điểm	Tỉ lệ	Điểm	Tỉ lệ	Điểm
<b>Yêu cầu:</b> Y1->Y6	25	5	125	5	125	5	125
Y7	7	1	7	5	35	5	35
Y8	7	1	7	5	35	5	35
Y9	8	5	40	4	32	5	40
Y10	3	1	3	1	3	5	15
	50		182		230		250
<b>Ràng buộc:</b>							
R1	10	5	50	5	50	5	50
R2	6	5	30	5	30	5	30
R3	6	5	30	5	30	5	30
R4	6	5	30	5	30	5	30
R5	4	5	20	4	16	1	4
R6	4	5	20	4	16	2	8
R7	4	5	20	4	16	2	8
R8	10	5	50	5	50	5	50
	50		200		238		160
<b>Tổng</b>			382		468		410

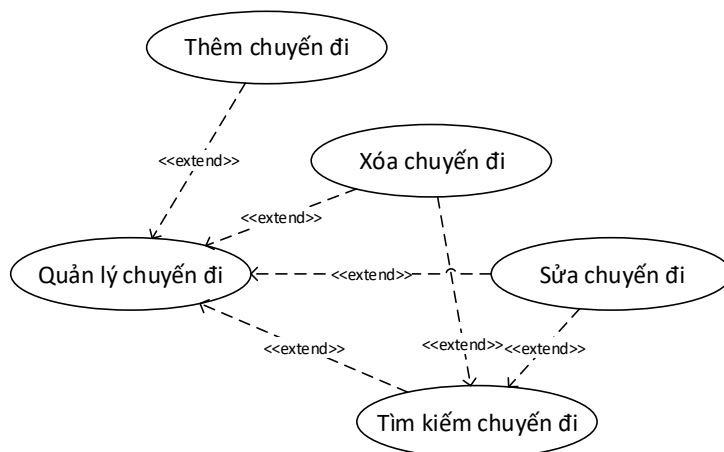
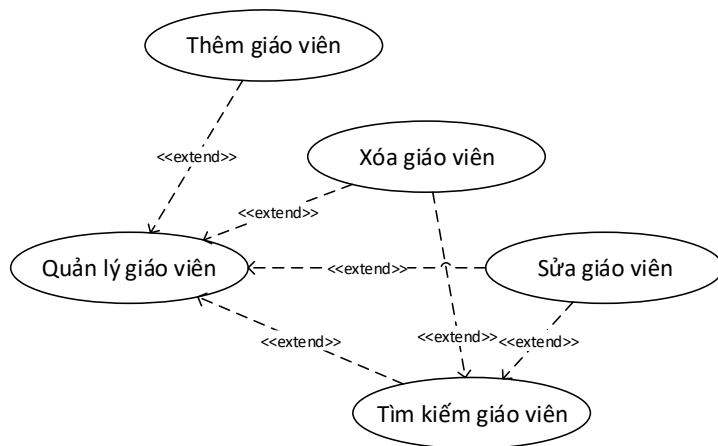
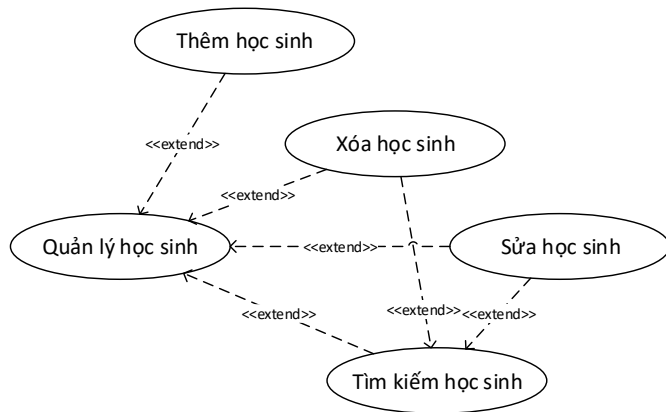
➔ Chọn phương án B

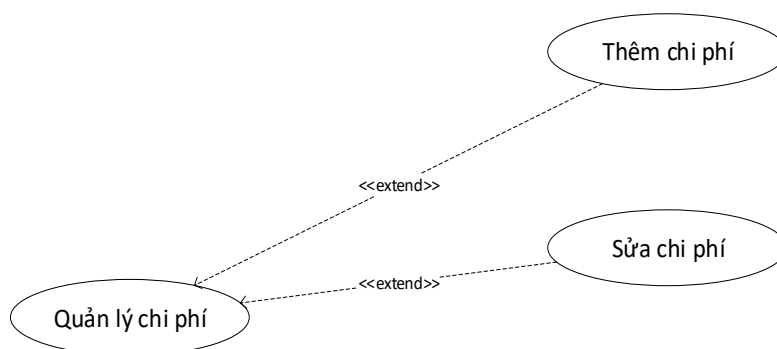
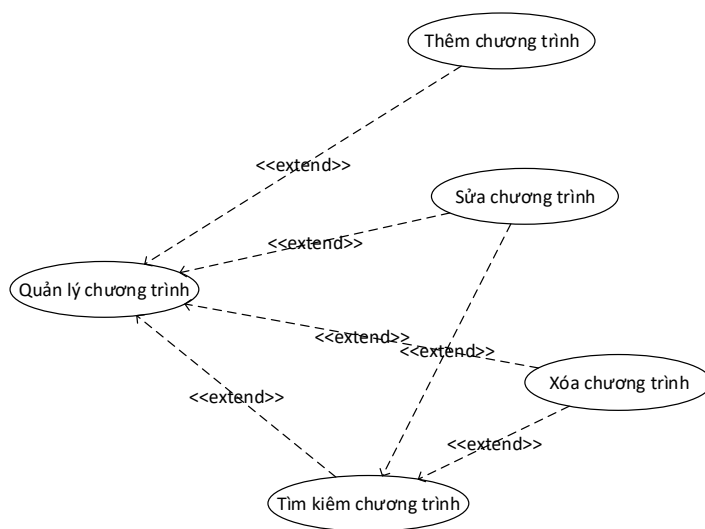
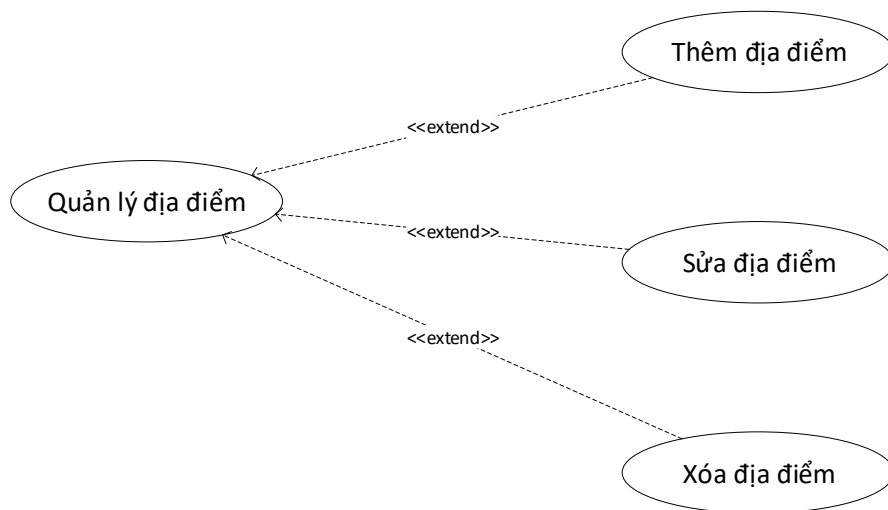
### 3. Thiết kế - đặc tả hệ thống

#### 3.1. Sơ đồ Usecase

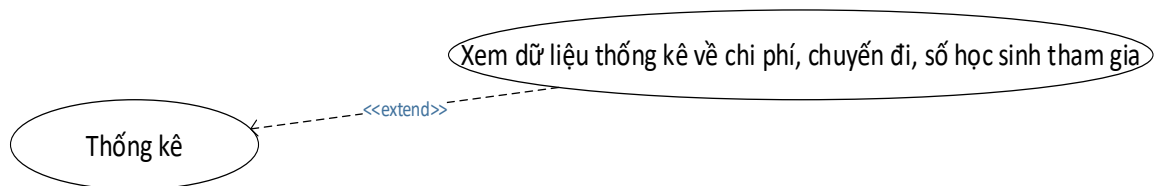
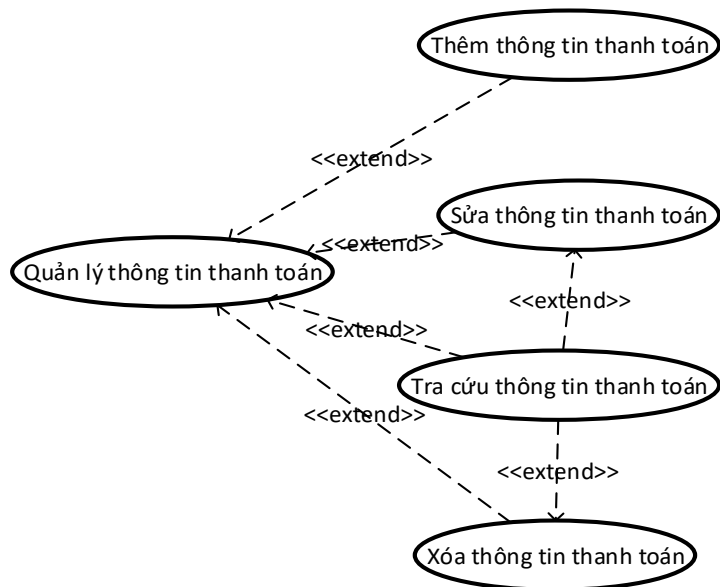
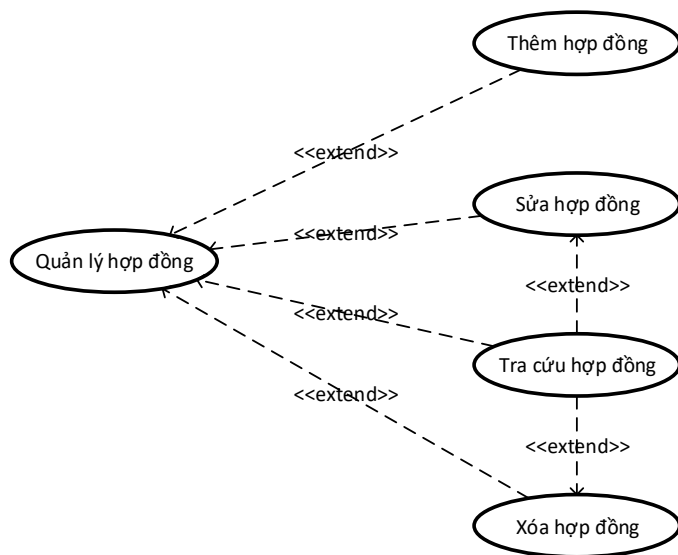


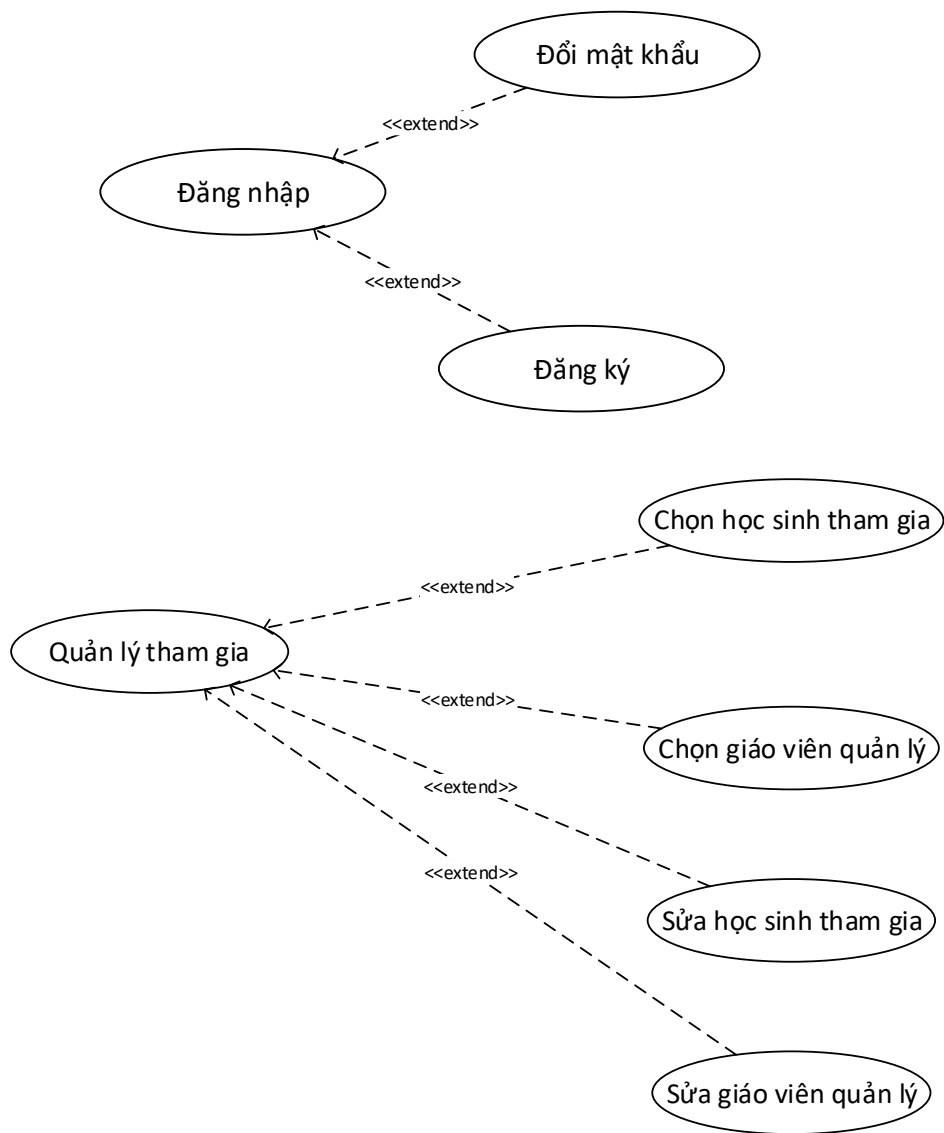
Trong đó gồm có:











### 3.2. Danh sách các tác nhân của hệ thống

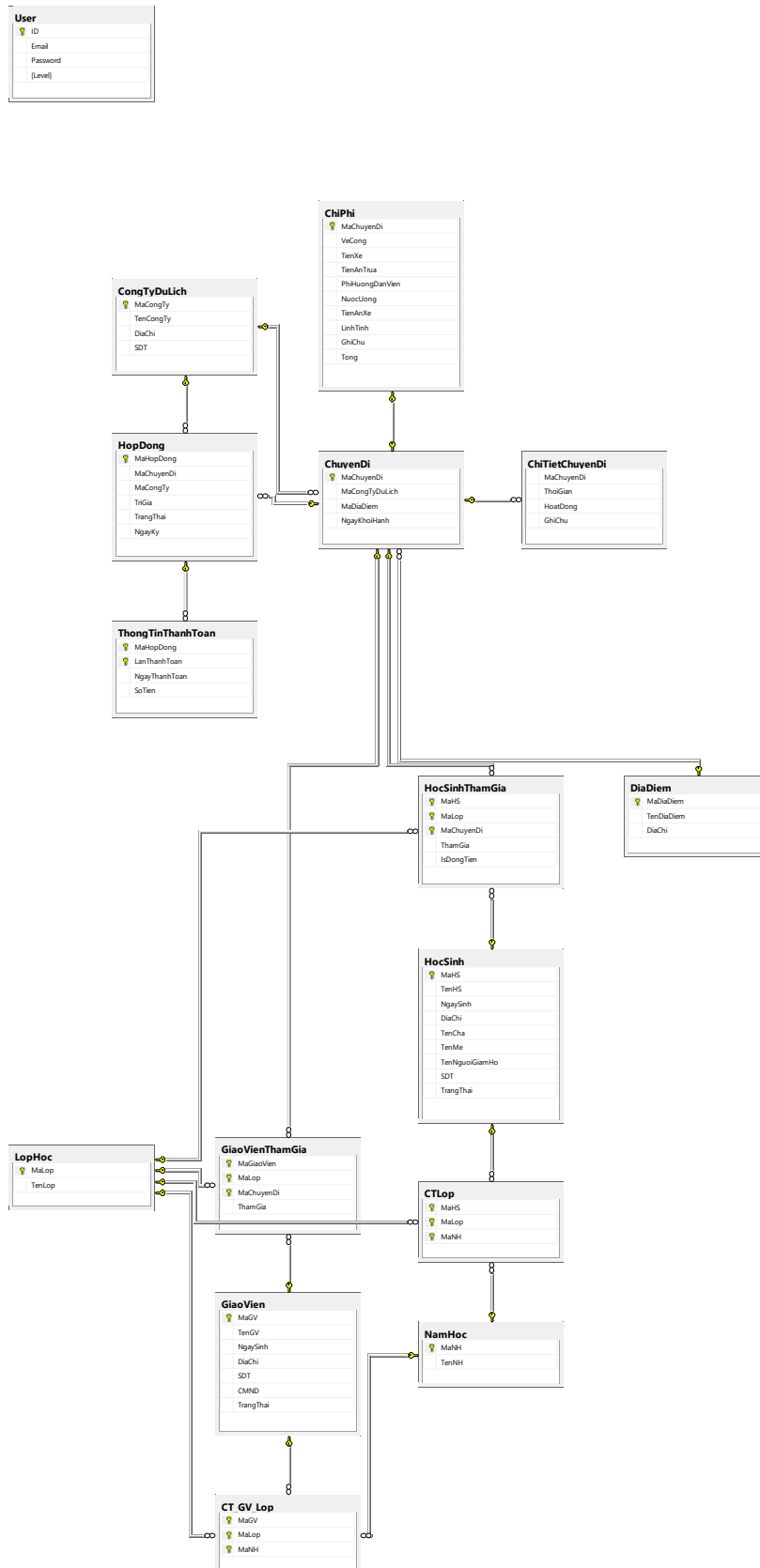
STT	Tác nhân	Ý nghĩa
1	Hiệu trưởng	Người sử dụng tất cả các chức năng của hệ thống. người quản trị hệ thống
2	Nhân viên	Người phụ giúp hiệu trưởng. Nhân viên chỉ dùng được một số chức năng nhất định trong hệ thống

### 3.3. Danh sách Usecase

STT	Usecase	Ý nghĩa
1	Quản lý học sinh	Thêm, xóa, sửa, tra cứu thông tin học sinh
2	Quản lý giáo viên	Thêm, xóa, sửa, tra cứu thông tin giáo viên
3	Quản lý chuyến đi	Thêm, xóa, sửa, tra cứu thông tin chuyến đi
4	Quản lý công ty du lịch	Thêm, xóa, sửa, tra cứu thông tin công ty du lịch
5	Quản lý địa điểm	Thêm, xóa, sửa thông tin địa điểm
6	Quản lý chương trình	Thêm, xóa, sửa, tra cứu thông tin chương trình
7	Quản lý chi phí	Thêm, sửa thông tin chi phí
8	Quản lý hợp đồng	Thêm, xóa, sửa, tra cứu thông tin hợp đồng
9	Quản lý chi tiết hợp đồng	Thêm, xóa, sửa, tra cứu thông tin chi tiết hợp đồng (các điều khoản)
9	Báo cáo thống kê	Thống kê lại tổng chi phí, chi phí trung bình, số lượng trẻ tham gia trung bình, số chuyến đi trung bình theo từng năm
10	Đăng nhập	Cho phép người dùng đăng nhập, đăng ký hoặc đổi mật khẩu.
11	Quản lý tham gia	Cho phép người dùng xem thông tin của các học sinh, giáo viên tham gia chuyến đi đó

## 4. Thiết kế cơ sở dữ liệu

### 4.1. Sơ đồ lớp



#### 4.2. Danh sách các thành phần trong sơ đồ lớp

STT	Tên	Ý nghĩa	Ghi chú
1	HocSinh	Thông tin học sinh	
2	NamHoc	Thông tin các năm học	
3	LopHoc	Thông tin các lớp học	
4	CTLop	Mã học sinh học các lớp theo năm học	
5	GiaoVien	Thông tin giáo viên	
6	CongTyDuLich	Thông tin công ty du lịch	
7	DiaDiem	Thông tin địa điểm dã ngoại	
8	ChuyenDi	Thông tin chuyến đi	
9	ChiTietChuyenDi	Thông tin chi tiết về chuyến đi	
10	ChiPhi	Các chi phí cho chuyến đi	
11	HopDong	Thông tin hợp đồng	
12	ThongTinThanhToan	Thông tin thanh toán của hợp đồng	
13	HocSinhThamGia	Thông tin tham gia chuyến đi dã ngoại của học sinh	
14	GiaoVienThamGia	Thông tin tham gia chuyến đi dã ngoại của giáo viên	
15	[User]	Thông tin đăng nhập	
16	CT_GV_Lop	Giáo viên dạy lớp theo năm học	

### 4.3. Mô tả chi tiết các lớp đối tượng

#### 4.3.1. Lớp đối tượng HocSinh

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaHS	Khóa chính	Int	Khác null	Mã học sinh
2	TenHS		Nvarchar(50)		Tên học sinh
3	NgaySinh		Smalldatetime		Ngày tháng năm sinh
4	DiaChi		Nvarchar(50)		Địa chỉ
5	TenCha		Nvarchar(50)		Họ tên cha
6	TenMe		Nvarchar(50)		Họ tên mẹ
7	TenNguoiGiamHo		Nvarchar(50)		Họ tên người giám hộ
8	SĐT		Nvarchar(11)		Số điện thoại
9	TrangThai		Bit		Trạng thái học

#### 4.3.2. Lớp đối tượng NamHoc

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaNH	Khóa chính	Int	Khác null	Mã năm học
2	TenNH		Nvarchar(20)		Tên năm học

#### 4.3.3. Lớp đối tượng LopHoc

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaLop	Khóa chính	Nvarchar(6)	Khác null	Mã lớp học
2	TenLop		Nvarchar(20)		Tên lớp học

#### 4.3.4. Lớp đối tượng CTLop

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaHS	Khóa chính Khóa ngoại	Int	Khác null Quan hệ với lớp HocSinh	Mã học sinh
2	MaLop	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp LopHoc	Tên lớp học
3	MaNH	Khóa chính Khóa ngoại	Int	Khác null Quan hệ với lớp NamHoc	Mã năm học

#### 4.3.5. Lớp đối tượng GiaoVien

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaGV	Khóa chính	Int	Khác null	Mã giáo viên
2	TenGV		Nvarchar(50)		Tên giáo viên
3	NgaySinh		Smalldatetime		Ngày tháng năm sinh
4	DiaChi		Nvarchar(50)		Địa chỉ
5	SĐT		Nvarchar(11)		Số điện thoại
6	CMND		Nvarchar(12)		Chứng minh nhân dân

#### 4.3.6. Lớp đối tượng CongTyDuLich

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaCongTy	Khóa chính	Nvarchar(6)	Khác null	Mã công ty
2	TenCongTy		Nvarchar(50)		Tên công ty
3	DiaChi		Nvarchar(255)		Địa chỉ
4	SĐT		Nvarchar(11)		Số điện thoại

#### 4.3.7. Lớp đối tượng DiaDiem

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaDiaDiem	Khóa chính	Nvarchar(6)	Khác null	Mã địa điểm
2	TenDiaDiem		Nvarchar(50)		Tên địa điểm
3	DiaChi		Nvarchar(255)		Địa chỉ

#### 4.3.8. Lớp đối tượng ChuyenDi

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaChuyenDi	Khóa chính	Nvarchar(6)	Khác null	Mã chuyến đi
2	MaCongTyDuLich	Khóa ngoại	Nvarchar(6)	Quan hệ với lớp CongTyDuLich	Mã công ty du lịch
3	MaDiaDiem	Khóa ngoại	Nvarchar(6)	Quan hệ với lớp DiaDiem	Mã địa điểm
4	NgayKhoiHanh		Date		Ngày khởi hành



#### 4.3.9. Lớp đối tượng ChiTietChuyenDi

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaChuyenDi	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp ChuyenDi	Mã chuyến đi
2	ThoiGian		Time(5)		Thời gian
3	HoatDong		Nvarchar(255)		Hoạt động
4	GhiChu		Nvarchar(255)		Ghi chú

#### 4.3.10. Lớp đối tượng ChiPhi

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaChuyenDi	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp ChuyenDi	Mã chuyến đi
2	VeCong		Float		Vé công
3	TienXe		Float		Tiền xe
4	TienAnTrua		Float		Tiền ăn trưa
5	PhiHuongDanVien		Float		Phí hướng dẫn viên
6	NuocUong		Float		Nước uống
7	TienAnXe		Float		Tiền ăn xe
8	LinhTinh		Float		Linh tinh
9	GhiChu		Nvarchar(255)		Ghi chú
10	Tong		Float		Tổng tiền

#### 4.3.11. Lớp đối tượng HopDong

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaHopDong	Khóa chính	Nvarchar(6)	Khác null	Mã hợp đồng
2	MaChuyenDi	Khóa ngoại	Nvarchar(6)	Quan hệ với lớp ChuyenDi	Mã chuyển đi
3	MaCongTy	Khóa ngoại	Nvarchar(6)	Quan hệ với lớp CongTyDuLich	Mã công ty
4	TriGia		Float		Trị giá
5	NgayKy		Date		Ngày kí
6	TrangThai		Nvarchar(20)		Trạng thái

#### 4.3.12. Lớp đối tượng ThôngTinThanhToan

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaHopDong	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp HopDong	Mã hợp đồng
2	LanThanhToan	Khóa chính	Int	Khác null	Lần thanh toán
3	NgayThanhToan		Smalldatetime		Ngày thanh toán
4	SoTien		Float		Số Tiền

#### 4.3.13. Lớp đối tượng HocSinhThamGia

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
	MaHS	Khóa chính Khóa ngoại	Int	Khác null Quan hệ với lớp HocSinh	Mã học sinh
2	MaLop	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp LopHoc	Mã lớp học
3	MaChuyenDi	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp ChuyenDi	Mã chuyển đi
4	ThamGia		Bit		Tham gia hay không
5	IsDongTien		Bit		Đóng tiền hay chưa

#### 4.3.14. Lớp đối tượng GiaoVienThamGia

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaGiaoVien	Khóa chính Khóa ngoại	Int	Khác null Quan hệ với lớp GiaoVien	Mã giáo viên
2	MaLop	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp LopHoc	Mã lớp học
3	MaChuyenDi	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp ChuyenDi	Mã chuyển đi
4	ThamGia		Bit		Tham gia hay không

#### 4.3.15. Lớp đối tượng [User]

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	ID	Khóa chính	Int	Khác null	Mã đăng nhập
2	Email		Nvarchar(50)		Email đăng nhập
3	Password		Nvarchar(50)		Mật khẩu
4	Level		Int		Cấp độ người dùng

#### 4.3.16. Lớp đối tượng CT\_GV\_Lop

STT	Tên thuộc tính	Loại	Kiểu	Ràng buộc	Diễn giải
1	MaGV	Khóa chính Khóa ngoại		Khác null Quan hệ với lớp GiaoVien	Mã giáo viên
2	MaLop	Khóa chính Khóa ngoại	Nvarchar(6)	Khác null Quan hệ với lớp LopHoc	Mã lớp học
3	MaNH	Khóa chính Khóa ngoại	Int	Khác null Quan hệ với lớp NamHoc	Mã năm học

## 5. Xây dựng và xử lý thao tác

### 5.1. Trigger

#### 5.1.1. Danh sách các trigger trong phần mềm

- Danh sách trigger

STT	Tên trigger	Ý nghĩa
1	NamSinhHS	Trigger dùng để xác định độ tuổi của trẻ từ 3-5 tuổi mới nhận Nếu không thông báo lỗi cho người dùng
2	SiSoLop	Trigger này dùng để set sĩ số lớp, sĩ số lớp tối đa là 50
3	UTG_CapNhatTrangThai	Sau khi xóa thông tin thanh toán của 1 hợp đồng, tính tổng số tiền đã thanh toán <ul style="list-style-type: none"><li>- Nếu lớn hơn hoặc bằng trị giá của hợp đồng-&gt; Cập nhật trạng thái thành “Đã thanh toán”</li><li>- Ngược lại, cập nhật trạng thái hợp đồng thành “Chưa thanh toán”</li></ul>
4	UTG_CapNhatTrangThai1	Sau khi thêm, sửa thông tin thanh toán của 1 hợp đồng, tính tổng số tiền đã thanh toán <ul style="list-style-type: none"><li>- Nếu lớn hơn hoặc bằng trị giá của hợp đồng-&gt; Cập nhật trạng thái thành “Đã thanh toán”</li><li>- Ngược lại, cập nhật trạng thái hợp đồng thành “Chưa thanh toán”</li></ul>
5	UTG_CapNhatTriGiaHopDong	Sau khi cập nhật trị giá của 1 hợp đồng, tính tổng số tiền đã thanh toán <ul style="list-style-type: none"><li>- Nếu lớn hơn hoặc bằng trị giá của hợp đồng-&gt; Cập nhật trạng thái thành “Đã thanh toán”</li><li>- Ngược lại, cập nhật trạng thái hợp đồng thành “Chưa thanh toán”</li></ul>
6	UTG_ngayKy	<ul style="list-style-type: none"><li>- Ràng buộc ngày ký hợp đồng phải nhỏ hơn ngày khởi hành của chuyến đi cho hợp đồng đó.</li><li>- Khi thêm/ cập nhật hợp đồng, lấy thuộc tính “NgayKy” so sánh với “NgayKhoiHanh” của chuyến đi ứng với hợp đồng đó, nếu không thỏa ràng buộc trên thì sẽ rollback</li></ul>

7	UTG_UpdateNgayKhoiHanh	<ul style="list-style-type: none"> <li>- Ràng buộc ngày ký hợp đồng phải nhỏ hơn ngày khởi hành của chuyến đi cho hợp đồng đó.</li> <li>- Khi cập nhật chuyến đi, kiểm tra chuyến đi đó có hợp đồng chưa:</li> </ul> <p>+ Nếu có thì lấy thuộc tính “NgayKy” trong hợp đồng của chuyến đi đó, so sánh với “NgayKhoiHanh” của chuyến đi, nếu không thỏa ràng buộc trên thì sẽ rollback</p> <p>+ Nếu chưa thì cho update mà không cần kiểm tra</p>
8	UTG_ngayThanhToan	<ul style="list-style-type: none"> <li>- Ràng buộc ngày thanh toán phải lớn hơn hoặc bằng ngày ký hợp đồng.</li> <li>- Khi thêm/ cập nhật thông tin thanh toán, lấy giá trị “NgayThanhToan” so sánh với “NgayKy” của hợp đồng. Nếu thỏa mãn ràng buộc trên thì cho phép thêm/ cập nhật, nếu không sẽ rollback.</li> </ul>
9	UTG_updateNgayKy	<ul style="list-style-type: none"> <li>- Ràng buộc ngày thanh toán phải lớn hơn hoặc bằng ngày ký hợp đồng.</li> <li>- Khi sửa hợp đồng, lấy giá trị “NgayKy” so sánh với “NgayThanhToan” của tất cả thông tin thanh toán của hợp đồng này. Nếu thỏa mãn ràng buộc trên thì cho phép sửa, nếu không sẽ rollback</li> </ul>
10	UTG_congTy_name	<ul style="list-style-type: none"> <li>- Ràng buộc tên công ty không được trùng</li> <li>- Khi thêm/ cập nhật công ty, so sánh tên công ty với tất cả công ty có trong CDSL, nếu trùng sẽ rollback</li> </ul>
11	UTG_congTy_SDT	<ul style="list-style-type: none"> <li>- Ràng buộc SĐT của công ty không được trùng</li> <li>- Khi thêm/ cập nhật công ty, so sánh SĐT với tất cả công ty có trong CDSL, nếu trùng sẽ rollback</li> </ul>
12	UTG_GiaoVien_CMND	<ul style="list-style-type: none"> <li>- Ràng buộc CMND của giáo viên không được trùng</li> <li>- Khi thêm/ cập nhật giáo viên, so sánh CMND với tất cả giáo viên có trong CDSL, nếu trùng sẽ rollback</li> </ul>

13	UTG_GiaoVien_SDT	<ul style="list-style-type: none"> <li>- Ràng buộc SĐT của giáo viên không được trùng nhau</li> <li>- Khi thêm/ cập nhật giáo viên, so sánh SĐT với tất cả giáo viên có trong CDSL, nếu trùng sẽ rollback</li> </ul>
14	UTG_HopDong_MaChuyenDi	<ul style="list-style-type: none"> <li>- Ràng buộc mã chuyển đi trong hợp đồng không được trùng nhau</li> <li>- Khi thêm/ cập nhật hợp đồng, kiểm tra “MaChuyenDi” có trùng với tất cả hợp đồng còn lại không, nếu có sẽ rollback</li> </ul>
15	checkTenDiaDiem_DiaDiem	Trigger checkTenDiaDiem_DiaDiem để check xem tên địa điểm lưu có bị trùng hay không
16	checkDiaChi_DiaDiem	Trigger checkDiaChi_DiaDiem để check xem địa chỉ địa điểm lưu có bị trùng hay không
17	checkTongTien_ChiPhi	Trigger checkTongTien_ChiPhi để check xem tổng chi phí có cao hơn 100 triệu đồng không

## 5.1.2. Mô tả các trigger

### - NamSinhHS

```
CREATE TRIGGER NamSinhHS
ON HocSinh
AFTER INSERT
AS
BEGIN
    DECLARE @MaHS INT, @NgaySinh SMALLDATETIME, @Now SMALLDATETIME
    SELECT @MaHS = MaHS, @NgaySinh = NgaySinh FROM Inserted
    IF( YEAR(@NgaySinh) < (YEAR(GETDATE()) - 5) or YEAR(@NgaySinh) >
        (YEAR(GETDATE()) - 3))
    BEGIN
        RAISERROR( 'Cannot insert HocSinh', 16, 1)
        ROLLBACK TRAN
    END
END
```

### - SiSoLop

```
CREATE TRIGGER SiSoLop
ON dbo.CTLop
AFTER INSERT
AS
BEGIN
    DECLARE @MaHS INT, @MaLop NVARCHAR(6), @MaNH INT, @tong int
    SELECT @MaLop = MaLop, @MaNH = MaNH FROM Inserted
    SELECT @tong = COUNT(MaHS) FROM dbo.CTLop WHERE MaNH = @MaNH AND MaLop = @MaLop
    IF( @tong > 50)
    BEGIN
        RAISERROR( 'Si so lop toi da, khong the them hoc sinh vao lop', 16, 1)
        ROLLBACK TRANSACTION
    END
END
```

### - UTG\_CapNhatTrangThai

```
CREATE TRIGGER UTG_CapNhatTrangThai
ON dbo.ThongTinThanhToan AFTER DELETE
AS
BEGIN
    DECLARE @maHD NVARCHAR(6)
    DECLARE @sum FLOAT
    DECLARE @triGiaHD FLOAT
    SELECT @maHD=Deleted.MaHopDong FROM Deleted
    SELECT @sum=SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=@maHD
    SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = @maHD
    IF(@sum >= @triGiaHD)
        UPDATE dbo.HopDong SET TrangThai = N'Đã thanh toán' WHERE
        MaHopDong=@maHD
    ELSE
        UPDATE dbo.HopDong SET TrangThai = N'Chưa thanh toán' WHERE
        MaHopDong=@maHD
END
```



- UTG\_CapNhatTrangThai1

```
CREATE TRIGGER UTG_CapNhatTrangThai1
ON dbo.ThongTinThanhToan AFTER INSERT,UPDATE
AS
BEGIN
    DECLARE @maHD NVARCHAR(6)
    DECLARE @sum FLOAT
    DECLARE @triGiaHD FLOAT
    SELECT @maHD=Inserted.MaHopDong FROM Inserted
    SELECT @sum=SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=@maHD
    SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = @maHD
    IF(@sum >= @triGiaHD)
        UPDATE dbo.HopDong SET TrangThai = N'Đã thanh toán' WHERE
MaHopDong=@maHD
    ELSE
        UPDATE dbo.HopDong SET TrangThai = N'Chưa thanh toán' WHERE
MaHopDong=@maHD
END
```

- UTG\_CapNhatTriGiaHopDong

```
CREATE TRIGGER UTG_CapNhatTriGiaHopDong
ON dbo.HopDong AFTER UPDATE
AS
BEGIN
    DECLARE @maHD NVARCHAR(6)
    DECLARE @sum FLOAT
    DECLARE @triGiaHD FLOAT
    SELECT @maHD=Inserted.MaHopDong FROM Inserted
    SELECT @sum=SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=@maHD
    SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = @maHD
    IF(@sum >= @triGiaHD)
        UPDATE dbo.HopDong SET TrangThai = N'Đã thanh toán' WHERE
MaHopDong=@maHD
    ELSE
        UPDATE dbo.HopDong SET TrangThai = N'Chưa thanh toán' WHERE
MaHopDong=@maHD
END
```

- UTG\_ngayKy

```
CREATE TRIGGER UTG_ngayKy ON dbo.HopDong
FOR INSERT,UPDATE
AS
BEGIN
    DECLARE @ngayKhoiHanh DATE
    DECLARE @ngayKy DATE
    SELECT @ngayKhoiHanh=dbo.ChuyenDi.NgayKhoiHanh FROM Inserted,dbo.ChuyenDi WHERE
Inserted.MaChuyenDi = ChuyenDi.MaChuyenDi
    SELECT @ngayKy=Inserted.NgayKy FROM Inserted
    IF(@ngayKy >= @ngayKhoiHanh)
    BEGIN
        RAISERROR (N'Ngày ký hợp đồng phải nhỏ hơn ngày khởi hành!',16,2)
        ROLLBACK
    END
END
GO
```

- UTG\_UpdateNgàyKhaiHanh

```
CREATE TRIGGER UTG_UpdateNgàyKhaiHanh ON dbo.ChuyenDi
FOR UPDATE
AS
BEGIN
    DECLARE @ngayKhaiHanh DATE
    DECLARE @ngayKy DATE
    SELECT @ngayKhaiHanh=Inserted.NgayKhaiHanh FROM Inserted
    SELECT @ngayKy=NgayKy FROM Inserted,dbo.HopDong WHERE
Inserted.MaChuyenDi=HopDong.MaChuyenDi
    IF(@ngayKy >= @ngayKhaiHanh)
    BEGIN
        RAISERROR (N'Ngày khởi hành phải lớn ngày ký hợp đồng!',16,2)
        ROLLBACK
    END
END
```

- UTG\_ngayThanhToan

```
CREATE TRIGGER UTG_ngayThanhToan ON dbo.ThongTinThanhToan
FOR INSERT,UPDATE
AS
BEGIN
    DECLARE @ngayThanhToan DATE
    DECLARE @ngayKy DATE
    SELECT @ngayThanhToan=Inserted.NgayThanhToan FROM Inserted
    SELECT @ngayKy=NgayKy FROM dbo.HopDong,Inserted WHERE HopDong.MaHopDong =
Inserted.MaHopDong
    IF(@ngayThanhToan<@ngayKy)
    BEGIN
        RAISERROR (N'Ngày thanh toán phải lớn hơn hoặc bằng ngày ký hợp
đồng.',16,1)
        ROLLBACK
    END
END
```

- UTG\_updateNgayKy

```
CREATE TRIGGER UTG_updateNgayKy ON dbo.HopDong
FOR UPDATE
AS
BEGIN
    DECLARE @maHopDong NVARCHAR(6)
    DECLARE @ngayKy DATE

    SELECT @maHopDong=Inserted.MaHopDong FROM Inserted
    SELECT @ngayKy = Inserted.NgayKy FROM Inserted
    DECLARE cs_ThanhToan CURSOR FOR SELECT NgayThanhToan FROM dbo.ThongTinThanhToan
WHERE MaHopDong=@maHopDong
    OPEN cs_ThanhToan
    DECLARE @ngayThanhToan DATE

    FETCH NEXT FROM cs_ThanhToan INTO @ngayThanhToan
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF (@ngayThanhToan<@ngayKy)
        BEGIN
            RAISERROR (N'Ngày ký hợp đồng phải nhỏ hơn hoặc bằng ngày
thanh toán!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_ThanhToan INTO @ngayThanhToan
    END
    CLOSE cs_ThanhToan
    DEALLOCATE cs_ThanhToan
END
```

- UTG\_congTy\_name

```
CREATE TRIGGER UTG_congTy_name ON dbo.CongTyDuLich
FOR INSERT,UPDATE
AS
BEGIN

    DECLARE @tenCongTy1 NVARCHAR(50)
    SELECT @tenCongTy1=Inserted.TenCongTy FROM Inserted
    DECLARE cs_congTy CURSOR FOR SELECT TenCongTy FROM dbo.CongTyDuLich
    OPEN cs_congTy
    DECLARE @tenCongTy2 NVARCHAR(50)
    DECLARE @count INT
    SET @count=0
    FETCH NEXT FROM cs_congTy INTO @tenCongTy2
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF(@tenCongTy1=@tenCongTy2)
        BEGIN
            SET @count=@count+1
        END
        IF(@count>1)
        BEGIN
            RAISERROR(N'Không được trùng tên công ty!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_congTy INTO @tenCongTy2
    END
    CLOSE cs_congTy
    DEALLOCATE cs_congTy
END
```

- UTG\_congTy\_SDT

```
CREATE TRIGGER UTG_congTy_SDT ON dbo.CongTyDuLich
FOR INSERT,UPDATE
AS
BEGIN

    DECLARE @sdt1 NVARCHAR(11)
    SELECT @sdt1=Inserted.SDT FROM Inserted
    DECLARE cs_congTy CURSOR FOR SELECT SDT FROM dbo.CongTyDuLich
    OPEN cs_congTy
    DECLARE @sdt2 NVARCHAR(50)
    DECLARE @count INT
    SET @count=0
    FETCH NEXT FROM cs_congTy INTO @sdt2
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF(@sdt1=@sdt2)
        BEGIN
            SET @count=@count+1
        END
        IF(@count>1)
        BEGIN
            RAISERROR(N'Không được trùng SDT!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_congTy INTO @sdt2
    END
    CLOSE cs_congTy
    DEALLOCATE cs_congTy
END
```

- UTG\_GiaoVien\_CMND

```
CREATE TRIGGER UTG_GiaoVien_CMND ON dbo.GiaoVien
FOR INSERT,UPDATE
AS
BEGIN

    DECLARE @cmd1 NVARCHAR(12)
    SELECT @cmd1=Inserted.CMND FROM Inserted
    DECLARE cs_GiaoVien CURSOR FOR SELECT CMND FROM dbo.GiaoVien
    OPEN cs_GiaoVien
    DECLARE @cmd2 NVARCHAR(12)
    DECLARE @count INT
    SET @count=0
    FETCH NEXT FROM cs_GiaoVien INTO @cmd2
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF (@cmd1=@cmd2)
        BEGIN
            SET @count=@count+1
        END
        IF (@count>1)
        BEGIN
            RAISERROR(N'Không được trùng CMND!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_GiaoVien INTO @cmd2
    END
    CLOSE cs_GiaoVien
    DEALLOCATE cs_GiaoVien
END
```

- UTG\_GiaoVien\_SDT

```
CREATE TRIGGER UTG_GiaoVien_SDT ON dbo.GiaoVien
FOR INSERT,UPDATE
AS
BEGIN

    DECLARE @sdt1 NVARCHAR(11)
    SELECT @sdt1=Inserted.SDT FROM Inserted
    DECLARE cs_GiaoVien CURSOR FOR SELECT SDT FROM dbo.GiaoVien
    OPEN cs_GiaoVien
    DECLARE @sdt2 NVARCHAR(12)
    DECLARE @count INT
    SET @count=0
    FETCH NEXT FROM cs_GiaoVien INTO @sdt2
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF(@sdt1=@sdt2)
        BEGIN
            SET @count=@count+1
        END
        IF(@count>1)
        BEGIN
            RAISERROR(N'Không được trùng số điện thoại!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_GiaoVien INTO @sdt2
    END
    CLOSE cs_GiaoVien
    DEALLOCATE cs_GiaoVien
END
```

- UTG\_HopDong\_MaChuyenDi

```
CREATE TRIGGER UTG_HopDong_MaChuyenDi ON dbo.HopDong
FOR INSERT,UPDATE
AS
BEGIN

    DECLARE @maChuyenDi1 NVARCHAR(6)
    SELECT @maChuyenDi1=Inserted.MaChuyenDi FROM Inserted
    DECLARE cs_HopDong CURSOR FOR SELECT MaChuyenDi FROM dbo.HopDong
    OPEN cs_HopDong
    DECLARE @maChuyenDi2 NVARCHAR(6)
    DECLARE @count INT
    SET @count=0
    FETCH NEXT FROM cs_HopDong INTO @maChuyenDi2
    WHILE @@FETCH_STATUS=0
    BEGIN
        IF (@maChuyenDi1=@maChuyenDi2)
        BEGIN
            SET @count=@count+1
        END
        IF (@count>1)
        BEGIN
            RAISERROR(N'Không được trùng chuyển đi!',16,1)
            ROLLBACK
        END
        FETCH NEXT FROM cs_HopDong INTO @maChuyenDi2
    END
    CLOSE cs_HopDong
    DEALLOCATE cs_HopDong
END
```



- checkTenDiaDiem\_DiaDiem

```
create trigger checkTenDiaDiem_DiaDiem
on DiaDiem
for insert, update
as
begin
    declare @tendd nvarchar(50)
    declare @tam nvarchar(50)
    declare @count int
    set @count=0
    select @tendd=TenDiaDiem from inserted
    declare csCheck cursor for
    select TenDiaDiem from DiaDiem
    open csCheck
    fetch next from csCheck
    into @tam
    while @@FETCH_STATUS = 0
    begin
        if @tendd=@tam
        begin
            set @count+=1
        end
        fetch next from csCheck
        into @tam
    end
    close csCheck
    deallocate cscheck
    if (@count >1)
    begin
        print N'Tên địa điểm không được trùng!!'
        rollback tran
    end
end
```

- checkDiaChi\_DiaDiem

```
create trigger checkDiaChi_DiaDiem
on DiaDiem
for insert, update
as
begin
    declare @diachi nvarchar(255)
    declare @tam nvarchar(255)
    declare @count int
    set @count=0
    select @diachi=DiaChi from inserted
    declare csCheck cursor for
    select DiaChi from DiaDiem
    open csCheck
    fetch next from csCheck
    into @tam
    while @@FETCH_STATUS = 0
    begin
        if @diachi=@tam
        begin
            set @count+=1
        end
        fetch next from csCheck
        into @tam
    end
    close csCheck
    deallocate cscheck
    if (@count >1)
    begin
        print N'Địa chỉ không được trùng!!'
        rollback tran
    end
end
```

- checkTongTien\_ChiPhi

```
create trigger checkTongTien_ChiPhi
on ChiPhi
for insert, update
as
begin
    declare @tong float
    select @tong=Tong from inserted
    if (@tong>100000000)
    begin
        print N'Tổng chi phí không được lớn hơn 100 triệu đồng!!'
        rollback tran
    end
end
```

## 5.2. Store procedure

### 5.2.1. Danh sách các Store procedure

- Danh sách các Store procedure

STT	Tên Procedure	Ý nghĩa
1	SelectHocSinh @MaNH INT, @MaLop NVARCHAR(6)	Procedure chọn học sinh theo lớp, theo năm học
2	SelectLopHoc	Lấy danh sách lớp học.
3	SelectNamHoc	Lấy danh sách năm học
4	ThemThamGia @MaNH int, @MaChuyenDi NVARCHAR(6)	Tự động thêm danh sách học sinh vào bảng học sinh tham gia sau khi thêm mỗi chuyến đi
5	ThemThamGiaGV @MaNH int, @MaChuyenDi NVARCHAR(6)	Tự động thêm danh sách giáo viên vào bảng giáo viên tham gia sau khi thêm mỗi chuyến đi
6	TimMaHS @MaHS INT OUT	Tìm mã học sinh phù hợp . Nếu học sinh chưa có mã số theo thứ tự thì thêm vào số thứ tự trống. Ngược lại sẽ thêm mã học sinh lớn nhất +1
7	CREATE PROC InsertHocSinh @TenHS NVARCHAR(50), @NgaySinh SMALLDATETIME, @DiaChi NVARCHAR(50), @TenCha NVARCHAR(50), @TenMe NVARCHAR(50), @TenNguoiGiamHo NVARCHAR(50), @SDT NVARCHAR(10), @MaLop NVARCHAR(6), @MaNH INT	Thêm học sinh, nếu học sinh bị trùng → thông báo lỗi Nếu học sinh không bị trùng → thêm mã học sinh , thêm học sinh vào cơ sở dữ liệu
8	TimMaUser @MaUser INT OUT	Tìm mã User phù hợp . Nếu User chưa có mã số theo thứ tự thì thêm vào số thứ tự trống. Ngược lại sẽ thêm mã User lớn nhất +1

9	InsertUser @Email NVARCHAR(50), @Pass NVARCHAR(50)	Thêm User, nếu email bị trùng → thông báo lỗi Nếu User không bị trùng → thêm mã User, thêm User vào cơ sở dữ liệu
10	USP_DeleteHopDong @maHD NVARCHAR(6)	Trước khi xóa hợp đồng phải xóa thông tin thanh toán của hợp đồng đó
11	create proc LuuChiPhi ( @MaChuyenDi nvarchar(6), @VeCong float, @TienXe float, @TienAnTrua float, @PhiHDV float, @NuocUong float, @TienAnXe float, @LinhTinh float, @GhiChu ntext )	Lưu chi phí chuyến đi
12	create proc chartTongChiPhi( @nam int)	Lấy dữ liệu thống kê chi phí theo từng năm
13	create proc chartChiPhiTB( @nam int)	Lấy dữ liệu thống kê chi phí trung bình theo từng năm
14	chartHSTB (@nam int)	chartHSTB thống kê số học sinh đi dã ngoại trung bình the từng năm
15	chartSoChuyenDi (@nam int)	Thống kê số chuyến đi trogn một năm

## 5.2.2. Mô tả các Store procedure

- SelectHocSinh @MaNH INT, @MaLop NVARCHAR(6)

```
CREATE PROC SelectHocSinh @MaNH INT, @MaLop NVARCHAR(6)
AS
    BEGIN
        IF (@@ERROR = 0 AND @@ROWCOUNT = 0)
            SELECT * FROM dbo.CTLop INNER JOIN dbo.HocSinh ON HocSinh.MaHS =
            CTLop.MaHS
            WHERE dbo.CTLop.MaLop = @MaLop AND dbo.CTLop.MaNH = @MaNH AND
            dbo.HocSinh.TrangThai = 'true'
        ELSE
            RAISERROR( 'Khong the hien thi danh sach hoc sinh', 16, 1)

        END

        SELECT * FROM dbo.CTLop INNER JOIN dbo.HocSinh ON HocSinh.MaHS = CTLop.MaHS
        WHERE dbo.CTLop.MaLop = 'Choi1' AND dbo.CTLop.MaNH = 2018 AND
        dbo.HocSinh.TrangThai = 'true'
```

- SelectLopHoc

```
CREATE PROC SelectLopHoc
AS
    BEGIN
        IF (@@ERROR = 0)
            SELECT * FROM dbo.LopHoc
        ELSE
            RAISERROR( 'Khong the hien thi danh sach Lop hoc', 16, 1)

        END
```

- SelectNamHoc

```
CREATE PROC SelectNamHoc
AS
    BEGIN
        IF (@@ERROR = 0 AND @@ROWCOUNT = 0)
            SELECT * FROM dbo.NamHoc
        ELSE
            RAISERROR( 'Khong the hien thi danh sach nam hoc', 17, 1)

        END
```

- ThemThamGia @MaNH int, @MaChuyenDi NVARCHAR(6)

```
CREATE PROC ThemThamGia @MaNH int, @MaChuyenDi NVARCHAR(6)
AS
BEGIN
    DECLARE @MaHS INT , @MaLop NVARCHAR(6)
    DECLARE CURSORTHamGia CURSOR FOR
    (SELECT MaHS, MaLop FROM dbo.CTLop WHERE MaNH = @MaNH)
    OPEN CURSORTHamGia
    FETCH NEXT FROM CURSORTHamGia INTO @MaHS, @MaLop
    WHILE @@FETCH_STATUS = 0
    BEGIN
        INSERT INTO dbo.HocSinhThamGia
        ( MaHS ,
          MaLop ,
          MaChuyenDi ,
          ThamGia ,
          IsDongTien
        )
        VALUES ( @MaHS, -- MaHS - int
                  @MaLop , -- MaLop - nvarchar(6)
                  @MaChuyenDi , -- MaChuyenDi - int
                  NULL , -- ThamGia - bit
                  NULL -- IsDongTien - bit
                )
        FETCH NEXT FROM CURSORTHamGia INTO @MaHS, @MaLop
    END
    CLOSE CURSORTHamGia
    DEALLOCATE CURSORTHamGia
END
```

- ThemThamGiaGV @MaNH int, @MaChuyenDi NVARCHAR(6)

```
CREATE PROC ThemThamGiaGV @MaNH int, @MaChuyenDi NVARCHAR(6)
AS
BEGIN
    DECLARE @MaGV INT, @MaLop NVARCHAR(6)
    DECLARE CURSORThamGia CURSOR FOR
    (SELECT MaGV, MaLop FROM dbo.CT_GV_Lop WHERE MaNH = @MaNH)
    OPEN CURSORThamGia
    FETCH NEXT FROM CURSORThamGia INTO @MaGV, @MaLop
    WHILE @@FETCH_STATUS = 0
    BEGIN
        INSERT INTO dbo.GiaoVienThamGia
        ( MaGiaoVien ,
          MaLop ,
          MaChuyenDi ,
          ThamGia
        )
        VALUES ( @MaGV , -- MaGiaoVien - int
                  @MaLop , -- MaLop - nvarchar(6)
                  @MaChuyenDi , -- MaChuyenDi - int
                  NULL -- ThamGia - bit
                )
        FETCH NEXT FROM CURSORThamGia INTO @MaGV, @MaLop
    END
    CLOSE CURSORThamGia
    DEALLOCATE CURSORThamGia
END
```

- TimMaHS @MaHS INT OUT

```
CREATE PROC TimMaHS @MaHS INT OUT
AS
BEGIN
    DECLARE @max INT, @i INT
    SELECT @max = MAX(MaHS) FROM dbo.HocSinh
    SET @i = 1;
    SET @MaHS = 0;
    WHILE @i < @max
    BEGIN
        IF (@i = SOME(SELECT MaHS FROM dbo.HocSinh))
            SET @i += 1
        ELSE
            BEGIN
                SET @MaHS = @i
                BREAK
            END
        END
    END
    IF (@MaHS = 0)
    BEGIN
        SET @MaHS = @max + 1
    END
END
```

- CREATE PROC InsertHocSinh @TenHS NVARCHAR(50), @NgaySinh SMALLDATETIME, @DiaChi NVARCHAR(50), @TenCha NVARCHAR(50), @TenMe NVARCHAR(50), @TenNguoiGiamHo NVARCHAR(50), @SDT NVARCHAR(10), @MaLop NVARCHAR(6), @MaNH INT

```

CREATE PROC InsertHocSinh @TenHS NVARCHAR(50), @NgaySinh SMALLDATETIME, @DiaChi
NVARCHAR(50), @TenCha NVARCHAR(50), @TenMe NVARCHAR(50), @TenNguoiGiamHo NVARCHAR(50),
@SDT NVARCHAR(10), @MaLop NVARCHAR(6), @MaNH INT
AS
    BEGIN
        DECLARE @MaHS INT, @temp INT
        EXEC TimMaHS @MaHS OUT
        IF (@TenHS = SOME(SELECT TenHS FROM dbo.HocSinh))
            IF (@NgaySinh = SOME(SELECT NgaySinh FROM dbo.HocSinh))
                IF (@DiaChi = SOME(SELECT DiaChi FROM dbo.HocSinh))
                    IF (@TenCha = SOME(SELECT TenCha FROM dbo.HocSinh))
                        IF (@SDT = SOME(SELECT SDT FROM dbo.HocSinh))
                            SET @temp = 1
        IF (@temp != 0)
            RAISERROR( 'Hoc Sinh bi trung', 16, 1)
        ELSE
            INSERT INTO dbo.HocSinh( MaHS , TenHS , NgaySinh , DiaChi , TenCha
, TenMe , TenNguoiGiamHo , SDT , TrangThai)
            VALUES ( @MaHS, @TenHS, @NgaySinh, @DiaChi, @TenCha, @TenMe ,
@TenNguoiGiamHo, @SDT, 1)
            INSERT INTO dbo.CTLop VALUES ( @MaHS, @MaLop, @MaNH)

    END

```

- TimMaUser @MaUser INT OUT

```

BEGIN
    DECLARE @max INT, @i INT
    SELECT @max = MAX(ID) FROM dbo.[User]
    SET @i = 1;
    SET @MaUser = 0;
    WHILE @i < @max
        BEGIN
            IF (@i = SOME(SELECT ID FROM dbo.[User]))
                SET @i += 1
            ELSE
                BEGIN
                    SET @MaUser = @i
                    BREAK
                END
            END
        END
    IF (@MaUser = 0)
        BEGIN
            SET @MaUser = @max + 1
        END
    END
END

```



- InsertUser @Email NVARCHAR(50), @Pass NVARCHAR(50)

```
CREATE PROC InsertUser @Email NVARCHAR(50), @Pass NVARCHAR(50)
AS
BEGIN
    DECLARE @MaUser INT, @temp INT
    EXEC TimMaUser @MaUser OUT
    IF( @Email = SOME(SELECT Email FROM dbo.[User]))
        SET @temp = 1
    IF(@temp != 0)
        RAISERROR( 'User bi trung' , 16,1)
    ELSE
        INSERT INTO dbo.[User]
        VALUES (@MaUser, @Email, @Pass, 1)
END
```

- USP\_DeleteHopDong @maHD NVARCHAR(6)

```
CREATE PROC USP_DeleteHopDong
@maHD NVARCHAR(6)
AS
BEGIN
    DELETE dbo.ThongTinThanhToan WHERE MaHopDong= @maHD
    DELETE dbo.HopDong WHERE MaHopDong= @maHD
END
```

- create proc LuuChiPhi (@MaChuyenDi nvarchar(6),@VeCong float, @TienXe float, @TienAnTrua float,@PhiHDV float, @NuocUong float, @TienAnXe float, @LinhTinh float, @GhiChu ntext )

```
create proc LuuChiPhi (
@MaChuyenDi nvarchar(6),
@VeCong float,
@TienXe float,
@TienAnTrua float,
@PhiHDV float,
@NuocUong float,
@TienAnXe float,
@LinhTinh float,
@GhiChu ntext )
as
begin
    begin
        update ChiPhi
        set VeCong=@VeCong, TienXe=@TienXe,
        TienAnTrua=@TienAnTrua, PhiHuongDanVien=@PhiHDV, NuocUong=@NuocUong, TienAnXe=@TienAnXe, L
        inhTinh=@LinhTinh, GhiChu=@GhiChu
        where MaChuyenDi=@MaChuyenDi
    end
    begin
        update ChiPhi
        set Tong=VeCong+TienXe+TienAnTrua+TienAnXe+PhiHuongDanVien+NuocUong+LinhTinh
        where MaChuyenDi=@MaChuyenDi
    end
end
```

- create proc chartTongChiPhi( @nam int)

```
create proc chartTongChiPhi( @nam int)
as
begin
    select sum(Tong) as TongChiPhi
    from ChiPhi, ChuyenDi
    where ChiPhi.MaChuyenDi=ChuyenDi.MaChuyenDi and
    year(ChuyenDi.NgayKhoiHanh)=@nam
end
```

- create proc chartChiPhiTB( @nam int)

```
create proc chartChiPhiTB( @nam int)
as
begin
    select round(avg(Tong),0) as ChiPhiTB
    from ChiPhi, ChuyenDi
    where ChiPhi.MaChuyenDi=ChuyenDi.MaChuyenDi and
    year(ChuyenDi.NgayKhoiHanh)=@nam
end
```

- chartHSTB (@nam int)

```
create proc chartHSTB (@nam int)
as
begin
    declare @TB float
    declare @HS float
    declare @CD float
    set @HS = (select count(MaHS) from ChuyenDi,HocSinhThamGia where
ChuyenDi.MaChuyenDi=HocSinhThamGia.MaChuyenDi and year(ChuyenDi.NgayKhoiHanh) = @nam
and HocSinhThamGia.ThamGia=1)
    set @CD = (select count(distinct(ChuyenDi.MaChuyenDi)) from
ChuyenDi,HocSinhThamGia where ChuyenDi.MaChuyenDi=HocSinhThamGia.MaChuyenDi and
year(ChuyenDi.NgayKhoiHanh) = @nam and HocSinhThamGia.ThamGia=1)
    set @TB = @HS / @CD
    select ROUND(@TB,0)
end
```

- chartSoChuyenDi (@nam int)

```
create proc chartSoChuyenDi (@nam int)
as
begin
    select count(MaChuyenDi) from ChuyenDi where year(NgayKhoiHanh) = @nam
end
```

## 6. Xử lý truy xuất đồng thời

### 6.1. Các mức cô lập trong SQL Server

#### 6.1.1. Read Uncommitted

- Đặc điểm:
  - Không thiết lập Share lock trên những đơn vị dữ liệu cần đọc. Do đó không phải chờ khi đọc dữ liệu (kể cả khi dữ liệu đang bị lock bởi giao tác khác)
  - Vẫn tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác
- Ưu điểm:
  - Tốc độ xử lý nhanh
  - Không cản trở những giao tác khác thực hiện việc cập nhật dữ liệu
- Khuyết điểm: Có khả năng xảy ra mọi vấn đề xử lý đồng thời:
  - Dirty Read
  - Unrepeatable Reads
  - Phantoms
  - Lost update

#### 6.1.2. Read Committed

- Đặc điểm
  - Đây là mức cô lập mặc định của SQL Server
  - Tạo Share Lock trên đơn vị dữ liệu đọc và giải phóng ngay sau đó
  - Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock này được giữ đến hết giao tác
- Ưu điểm
  - Giải quyết vấn đề Dirty Reads
  - Share lock được giải phóng ngay nên không ảnh hưởng đến thao tác cập nhật của các giao tác khác
- Khuyết điểm: Chưa giải quyết được các vấn đề
  - Unrepeatable reads, Phantoms, LostUpdate
  - Giao tác khác phải chờ khi dữ liệu cần đọc đang được giữ khóa ghi

### 6.1.3. Repeated Read

- Đặc điểm
  - Tạo Share Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác → Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này
  - Repeatable Read = Read Committed + Giải quyết Unrepeatable Reads
  - Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
- Ưu điểm: giải quyết được vấn đề Dirty Reads và Unrepeatable Reads
- Khuyết điểm
  - Chưa giải quyết được vấn đề Phantoms, do vẫn cho phép insert những dòng dữ liệu thỏa điều kiện thiết lập shared lock
  - Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)
  - Shared lock được giữ đến hết giao tác → cản trở việc cập nhật dữ liệu của các giao tác khác

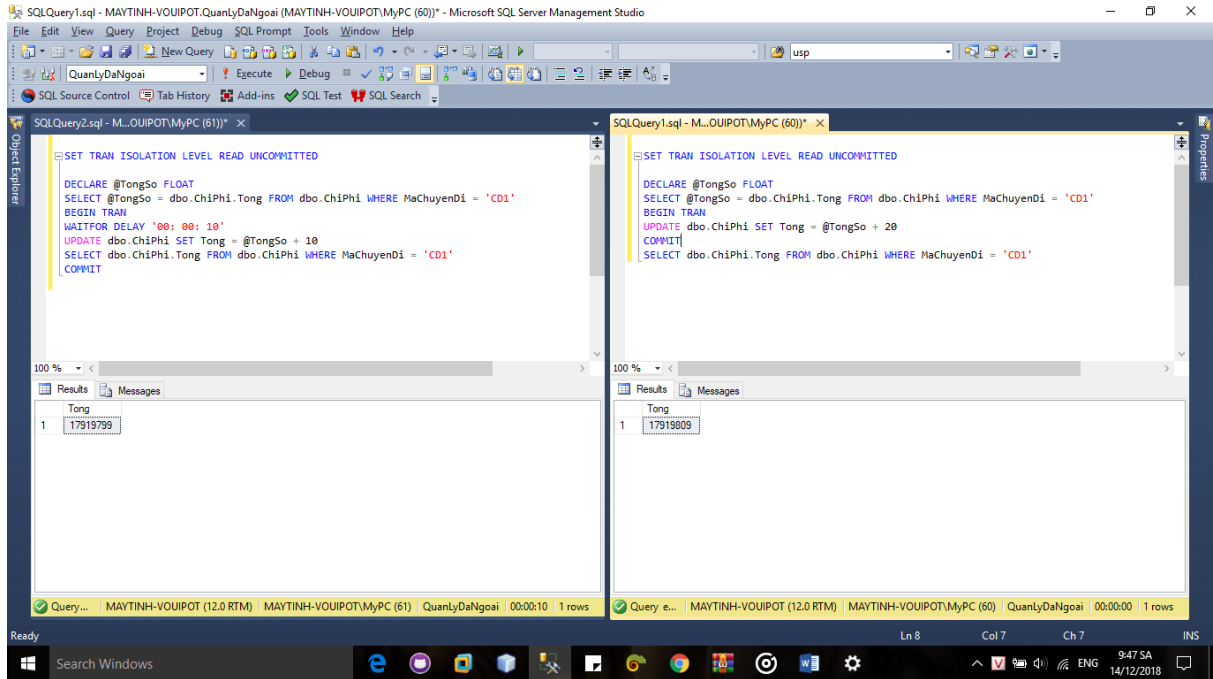
### 6.1.4. Serializable

- Đặc điểm:
  - Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác → Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này.
  - Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock) → Serializable = Repeatable Read + Giải quyết Phantoms
  - Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác
- Ưu điểm: Giải quyết được vấn đề Phantoms
- Khuyết điểm
  - Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)
  - Cản trở nhiều đến việc cập nhật dữ liệu của các giao tác khác

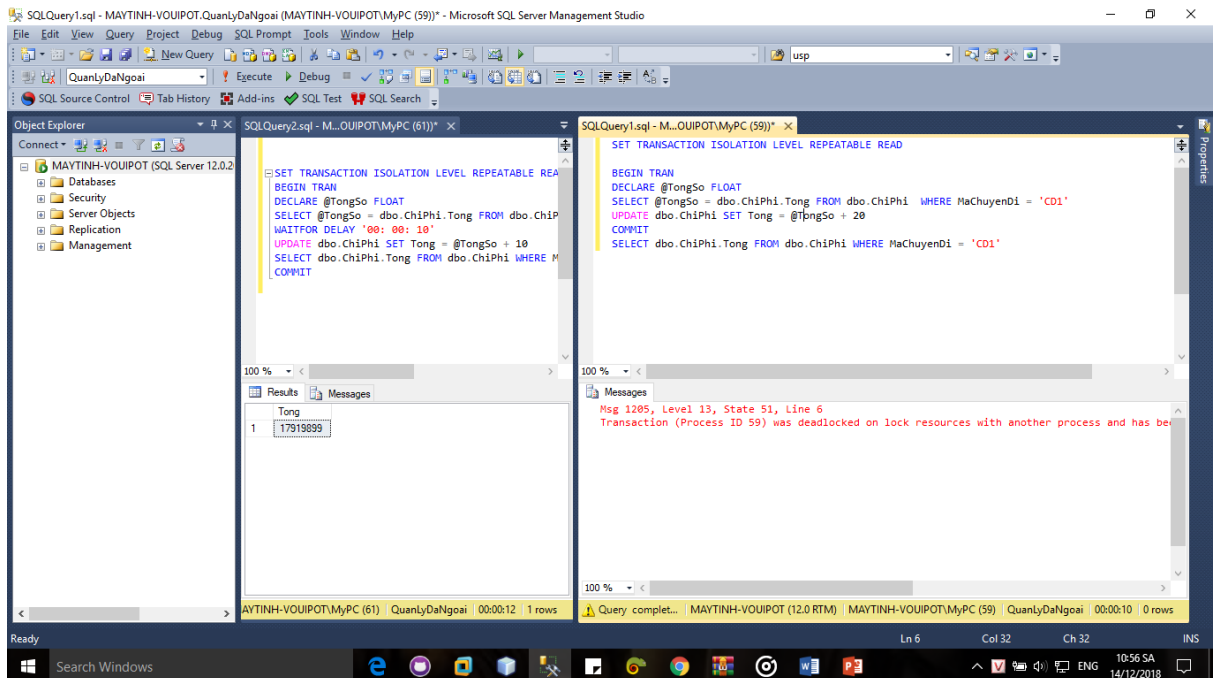
## 6.2. Các vấn đề xảy ra khi truy vấn đồng thời

### 6.2.1. Lost update

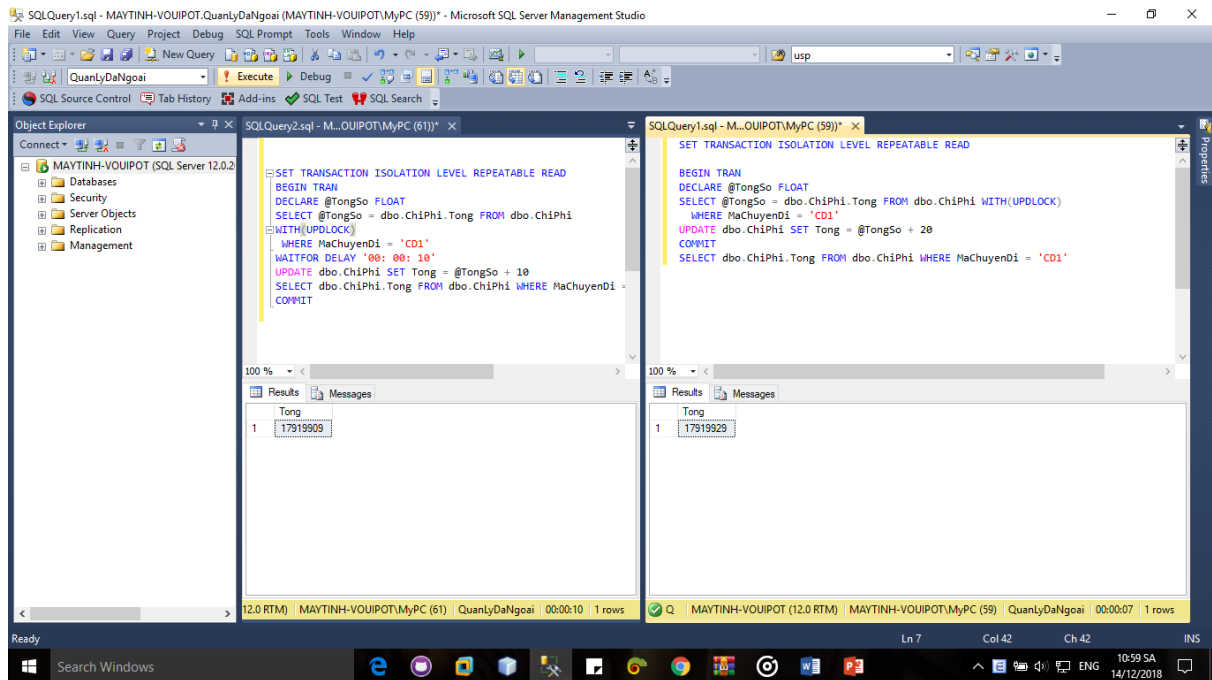
- ❖ Trường hợp 1: Khi cập nhật tổng số chi phí trong 1 chuyến đi, T1 cập nhật lên 10. T2 cập nhật lên 20. Giả sử T1 bị delay 10s. Thì T2 thực hiện update trước. sau đó T1 lấy lại giá trị cũ để cập nhật. Trong trường hợp này. Cả hai đều tăng chi phí của chuyến đi. Nhưng sau khi cập nhật. dẫn đến kết quả tổng sai. Mất dữ liệu cập nhật của T2



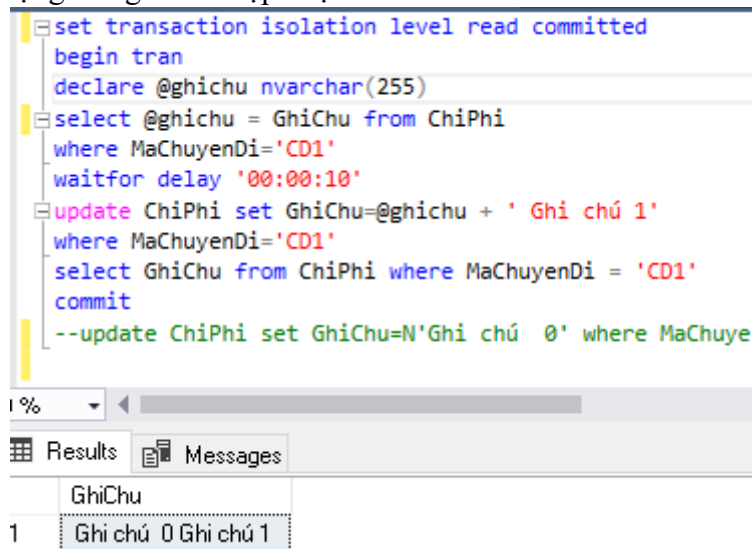
- Khi sử dụng chế độ Repeatable Read : xảy ra deadlock



- Khi chuyển đổi share lock thành update lock : giải quyết được deadlock



- ❖ Trường hợp 2: Khi cả 2 giao tác T1, T2 cùng sửa trên 1 đơn vị dữ liệu Ghi chú trong bảng Chi phí (T1 chưa commit) khiến cho tác dụng của giao tác cập nhật của T1 sẽ đè lên tác dụng của giao tác cập nhật T2



(Ghi chú ban đầu là “Ghi chú 0” sau đó T1 update thêm thành “Không Ghi chú 1”)

```

set transaction isolation level read committed
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi
where MaChuyenDi='CD1'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 2'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit

```

%

Results Messages

GhiChu
Ghi chú 0 Ghi chú 2

(Ghi chú ban đầu là “Không” sau đó T2 update thêm thành “Không Ghi chú 2”)

```

select GhiChu from ChiPhi where MaChuyenDi = 'CD1'

```

90 %

Results Messages

	GhiChu
1	Ghi chú 0 Ghi chú 1

(Kết quả là T1 đề dữ liệu lên T2)




- Giải quyết

- o Cách 1: Khi sử dụng Read uncommitted hoặc Read committed, ta chuyển thành Update lock, từ đó T2 sẽ phải chờ T1 hoàn tất để nhả khóa

```

set transaction isolation level read committed
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi with(updlock)
where MaChuyenDi='CD1'
waitfor delay '00:00:10'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 1'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit
--update ChiPhi set GhiChu=N'Ghi chú 0' where MaChuy

```

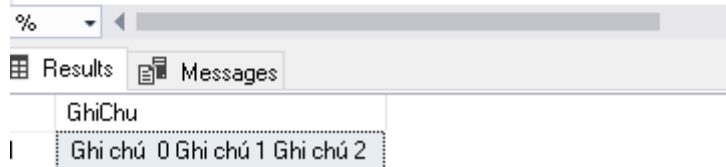


GhiChu
Ghi chú 0 Ghi chú 1

```

set transaction isolation level read committed
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi with(updlock)
where MaChuyenDi='CD1'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 2'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit

```



GhiChu
Ghi chú 0 Ghi chú 1 Ghi chú 2

(Thêm Update Lock => giải quyết được )

- Cách 2: Sử dụng Repeatable Read => T2 chờ T1 thực hiện xong mới thực hiện update
- Vấn đề: Xảy ra deadlock

SQLQuery2.sql - D:\...OMEACG\Tung (56))\*

```

set transaction isolation level repeatable read
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi
where MaChuyenDi='CD1'
waitfor delay '00:00:10'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 1'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit
--update ChiPhi set GhiChu=N'Ghi chú 0' where MaChuyenDi='CD

```

90 %

Results Messages

	GhiChu
1	Ghi chú 0 Ghi chú 1

SQLQuery1.sql - D:\...OMEACG\Tung (55))\*

```

set transaction isolation level repeatable read
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi
where MaChuyenDi='CD1'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 2'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit

```

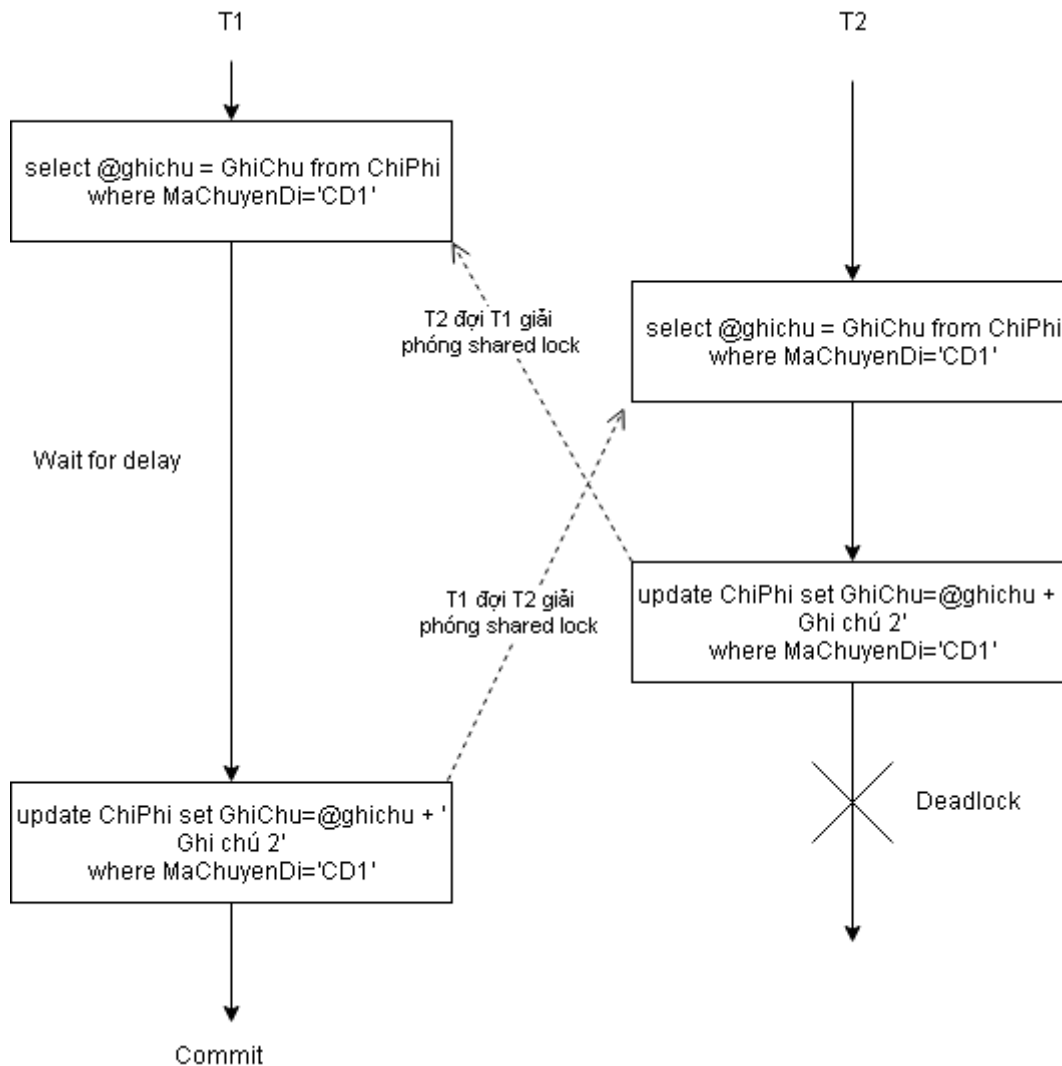
90 %

Messages

Msg 1205, Level 13, State 51, Line 6  
Transaction (Process ID 55) was deadlocked on lock resources with an

(Xảy ra deadlock)

- Nguyên nhân:



- **Giải quyết:** Thêm update lock vào lệnh select => T2 phải chờ T1 commit mới được thực hiện

```

set transaction isolation level repeatable read
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi with(updlock)
where MaChuyenDi='CD1'
waitfor delay '00:00:10'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 1'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit
--update ChiPhi set GhiChu=N'Ghi chú 0' where MaChuyenDi=
  
```

The screenshot shows a SQL Server query window with the above code. Below the code, the 'Results' tab is active, displaying a single row with the value 'Ghi chú 0 Ghi chú 1' for the GhiChu column.

	GhiChu
1	Ghi chú 0 Ghi chú 1

```

set transaction isolation level repeatable read
begin tran
declare @ghichu nvarchar(255)
select @ghichu = GhiChu from ChiPhi with(updlock)
where MaChuyenDi='CD1'
update ChiPhi set GhiChu=@ghichu + ' Ghi chú 2'
where MaChuyenDi='CD1'
select GhiChu from ChiPhi where MaChuyenDi = 'CD1'
commit

```

%

Results Messages

	GhiChu
1	Ghi chú 0 Ghi chú 1 Ghi chú 2

- ❖ Trường hợp 3: 2 nhân viên cùng update trị giá hợp đồng, thông tin của người này có thể chồng lên người kia  
⇒ Thông tin bị sai

```

--Lost update
--T1
SET TRAN ISOLATION LEVEL READ COMMITTED
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
WAITFOR DELAY '00: 00: 10'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 100 WHERE MaHopDong = 'HD1'
COMMIT

SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'

```

121 %

Results Messages

	TriGia
1	200100

```
--Lost update
--T2

SET TRAN ISOLATION LEVEL READ COMMITTED
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 200 WHERE MaHopDong = 'HD1'
COMMIT

SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
```

133 %

Results Messages

	TriGia
1	200200

(Giá trị ban đầu của trị giá hợp đồng HD1 là 200000)

- **Nguyên nhân:** T1 và T2 lấy giá trị hợp đồng. Khi T1 bị delay, T2 tiến hành update mà không biết rằng giá trị hợp đồng mới đã được T1 update, do đó T2 đang sử dụng giá trị cũ. Giá trị hợp đồng bị cập nhật sai.
- **Giải pháp:** Sử dụng mức cô lập repeatable read. Khi đó T2 phải đợi T1 kết thúc thì mới tiến hành update.

**Vấn đề:** Xảy ra deadlock

```
--Lost update
--T1

SET TRAN ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
WAITFOR DELAY '00: 00: 10'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 100 WHERE MaHopDong = 'HD1'
COMMIT

SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
```

%

Results Messages

	TriGia
	200100

```

Giải pháp nhưng de...LyDaNgoai (sa (54)) 1.sql - DESKTOP-BL...LyDaNgoai (sa (58)) 2.sql - DESKTOP-BL...LyDaNgoai (sa (51))
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 200 WHERE MaHopDong = 'HD1'
COMMIT

SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'

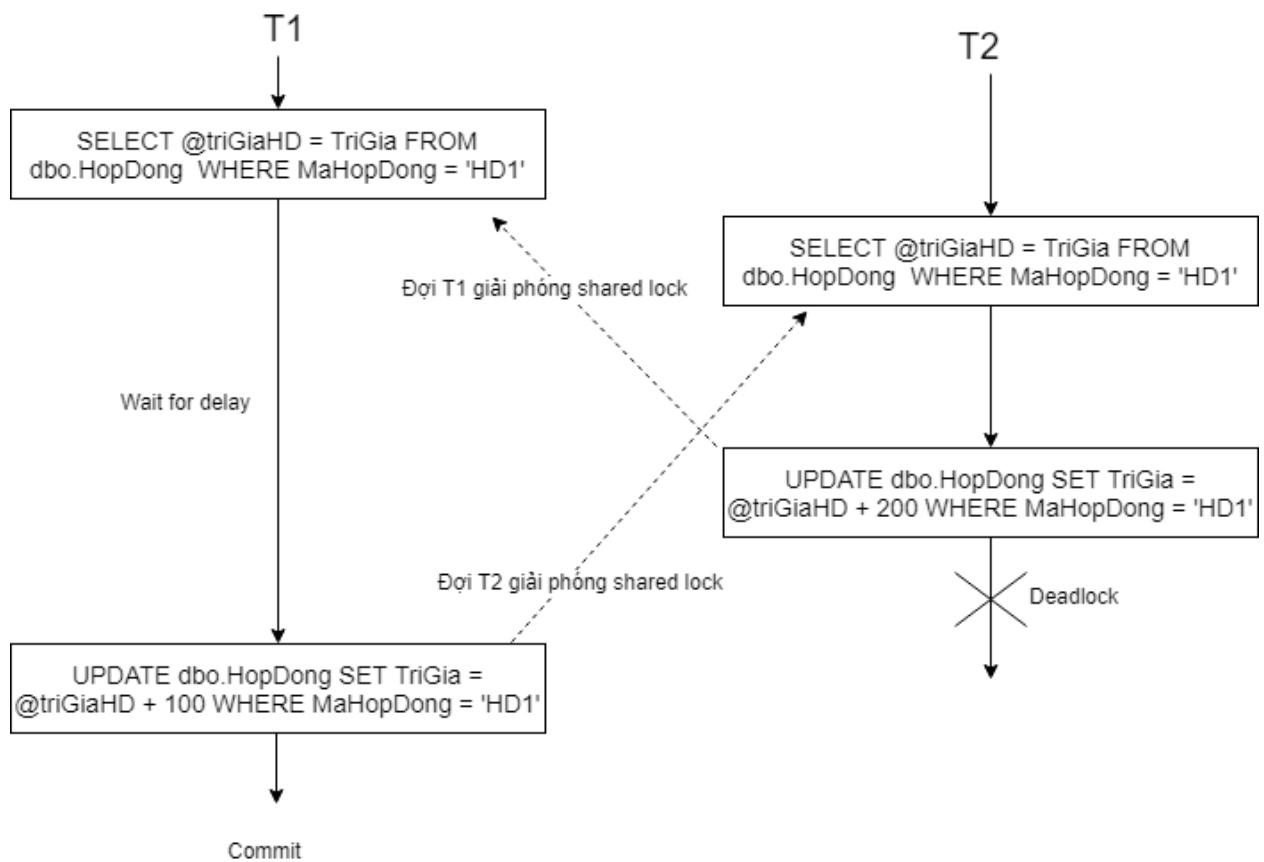
```

133 %

Messages

Msg 1205, Level 13, State 51, Line 5  
Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

- Nguyên nhân deadlock:



( Khi xảy ra deadlock, hệ quản trị sẽ tự động rollback transaction có kích cỡ nhỏ hơn. Kích cỡ của 1 transaction được xác định bởi số dòng insert, update or delete. Trong trường hợp này, 2 transaction có kích cỡ bằng nhau nên hệ quản trị sẽ rollback transaction T2 thực hiện sau.)

- Khắc phục deadlock: Thiết lập khóa Update Lock cho câu lệnh select

```
--Lost update
--T1

SET TRAN ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WITH (UPDLOCK) WHERE MaHopDong = 'HD1'
WAITFOR DELAY '00: 00: 10'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 100 WHERE MaHopDong = 'HD1'
COMMIT
|
SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'
```

121 %

Results Messages

	TriGia
1	200100

```
--Lost update
--T2

SET TRAN ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
DECLARE @triGiaHD FLOAT
SELECT @triGiaHD = TriGia FROM dbo.HopDong WITH (UPDLOCK) WHERE MaHopDong = 'HD1'
UPDATE dbo.HopDong SET TriGia = @triGiaHD + 200 WHERE MaHopDong = 'HD1'
COMMIT

SELECT TriGia FROM dbo.HopDong WHERE MaHopDong = 'HD1'|
```

133 %

Results Messages

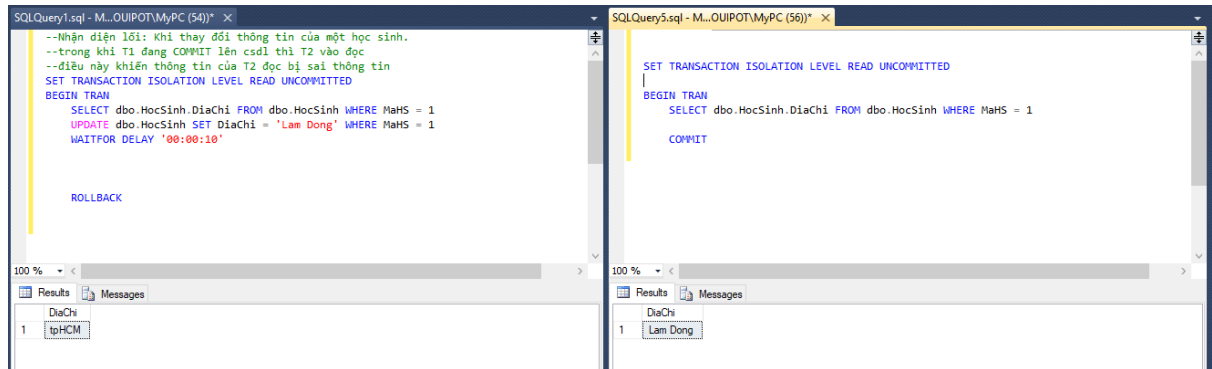
	TriGia
1	200300

(Giá trị ban đầu của trị giá hợp đồng HD1 là 200000)

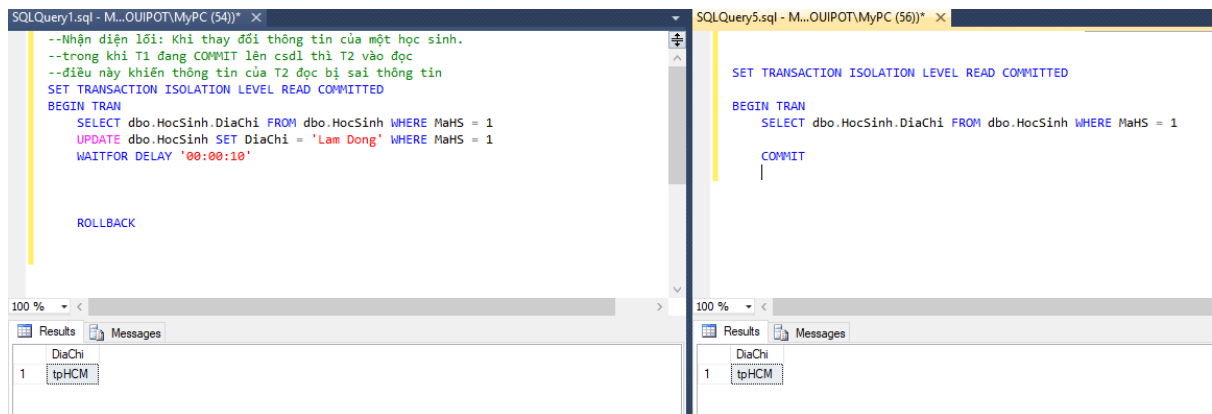
## 6.2.2. Dirty read

❖ Trường hợp 1:

- Khi T1 đang cập nhật dữ liệu thì T2 vào đọc.
- Tuy nhiên dữ liệu chưa được cập nhật xuống hệ thống. Vậy T2 đang đọc dữ liệu rác.
- Do mức cô lập Read Uncommitted không thiết lập share lock nên T2 vào đọc dữ liệu. Sau đó T1 rollback thì T2 không vào được nữa.



- Cách giải quyết: sử dụng mức cô lập Read Committed. Khi T1 thực hiện, tạo Exclusive lock nên giữ đến hết giao tác. Khi T2 vào, phải đợi đến khi T1 thực hiện xong mới đọc dữ liệu được





- ❖ Trường hợp 2: T1 đang xóa dữ liệu của chuyến đi CD10 thì T2 thực hiện đọc dữ liệu (đếm số chuyến đi trong năm 2018 có chuyến đi CD10). Lúc này T1 roll back nhưng T2 đã đếm sai số chuyến đi.

```
select MaChuyenDi from ChuyenDi
where year(NgayKhoiHanh)=2018
```

	MaChuyenDi
1	CD1
2	CD10
3	CD4

(Ban đầu có 3 chuyến đi)

```
begin tran
delete from ChuyenDi where MaChuyenDi='CD10'
waitfor delay '00:00:05'
rollback

select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
```

	(No column name)
1	3

(T1 roll back và đếm ra 3)

```
set transaction isolation level read uncommitted
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
```

	(No column name)
1	2

(T2 đếm sai ra 2)

**Nguyên nhân:** T2 đang là no lock nên k chờ T1 roll back mà đọc giá trị luôn

### Giải quyết:

- Cách 1: Chuyển mức cô lập của T2 về Read committed, tức tạo khóa shared lock cho T2, shared lock sau khi đọc xong dữ liệu được giải phóng nên không cản trở đến thao tác cập nhật của T1 => T2 chờ T1 thực hiện xong giao tác rồi mới đọc

```
begin tran
delete from ChuyenDi where MaChuyenDi='CD10'
waitfor delay '00:00:05'
rollback

select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
```

90 %

Results Messages

	(No column name)
1	3

```
set transaction isolation level read committed
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
```

90 %

Results Messages

	(No column name)
1	3

- Cách 2: Chuyển select của T2 từ No lock thành Update lock, T2 sẽ phải chờ T1 nhả khóa Exclusive lock => T2 phải chờ T1 hoàn tất mới thực hiện giao tác.

```
begin tran
delete from ChuyenDi where MaChuyenDi='CD10'
waitfor delay '00:00:05'
rollback

select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
```

10 %

Results Messages

	(No column name)
1	3

```

set transaction isolation level read uncommitted
select count(MaChuyenDi) from ChuyenDi with(updlock)
where year(NgayKhoiHanh)=2018

```

90 %

Results Messages

	(No column name)
1	3

- Trường hợp 3: Cô hiệu trưởng cần thông tin số điện thoại của một giáo viên. Trong lúc xem, một nhân viên tiến hành cập nhật lại SĐT của giáo viên này.

⇒ Cô hiệu trưởng gọi không được.

```

--Unrepeatable data
--T1
BEGIN TRAN
SELECT SĐT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
WAITFOR DELAY '00:00:10'
SELECT SĐT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT

```

177 %

Results Messages

	SĐT
1	11111

	SĐT
1	012345

```

--Unrepeatable data
--T2
BEGIN TRAN
UPDATE dbo.GiaoVien SET SĐT = N'012345' WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT

```

177 %

Messages

(1 row(s) affected)

- Nguyên nhân: Do T1 đang đọc giá trị , T2 vào ghi dữ liệu mới.
- Giải pháp: Chuyển mức cô lập của T1 thành repeatable read. Khi đó T2 phải đợi T1 kết thúc mới update được.

```
--Unrepeatable data
--T1
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
WAITFOR DELAY '00:00:05'
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT
```

161 %

Results Messages

	SDT
1	012345

	SDT
1	012345

- ❖ Trường hợp 3: Trong lúc một nhân viên đang update trạng thái của giáo viên có mã số 2 về 0( Không còn dạy) thì cô hiệu trưởng vào xem danh sách giáo viên đang giảng dạy thì nhân viên hủy giao tác.
  - ⇒ Danh sách cô hiệu trưởng xem ko chính xác

```
--Dirty reads
--T1|
BEGIN TRAN
UPDATE dbo.GiaoVien SET TrangThai =0 WHERE MaGV=2
WAITFOR DELAY '00:00:05'
ROLLBACK

SELECT * FROM dbo.GiaoVien WHERE TrangThai = 1
```

133 %

Results Messages

	MaGV	TenGV	NgaySinh	DiaChi	SDT	CMND	TrangThai
1	2	Phạm Quốc Cường	2018-12-20 00:00:00	Bình Dương	02165	5465132	1
2	3	Nguyễn Phạm Hương	2018-12-17 00:00:00	HCM	0231322	01123	1
3	4	Dương Văn Nam	2018-12-17 00:00:00	Đồng Nai	01123	032123	1

```
--Dirty reads

-- T2
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
SELECT * FROM dbo.GiaoVien WHERE TrangThai = 1
```

133 %

Results Messages

	MaGV	TenGV	NgaySinh	DiaChi	SDT	CMND	TrangThai
1	3	Nguyễn Phạm Hương	2018-12-17 00:00:00	HCM	0231322	01123	1
2	4	Dương Văn Nam	2018-12-17 00:00:00	Đồng Nai	01123	032123	1

- Nguyên nhân: T2 không được thiết lập shared lock nên đọc giá trị chưa commit
- Giải pháp: Chuyển mức cô lập của T2 về Read Committed (mức mặc định), vì mức cô lập này thiết lập Shared Lock trên đơn vị dữ liệu được đọc, Shared Lock được giải phóng ngay sau khi đọc xong dữ liệu. Do đó T2 phải chờ T1 kết thúc giao tác mới tiến hành đọc

```
--Giải quyết dirty reads
--Chuyển mức cô lập của T2 về Read Committed
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
SELECT * FROM dbo.GiaoVien WHERE TrangThai = 1
```

133 %

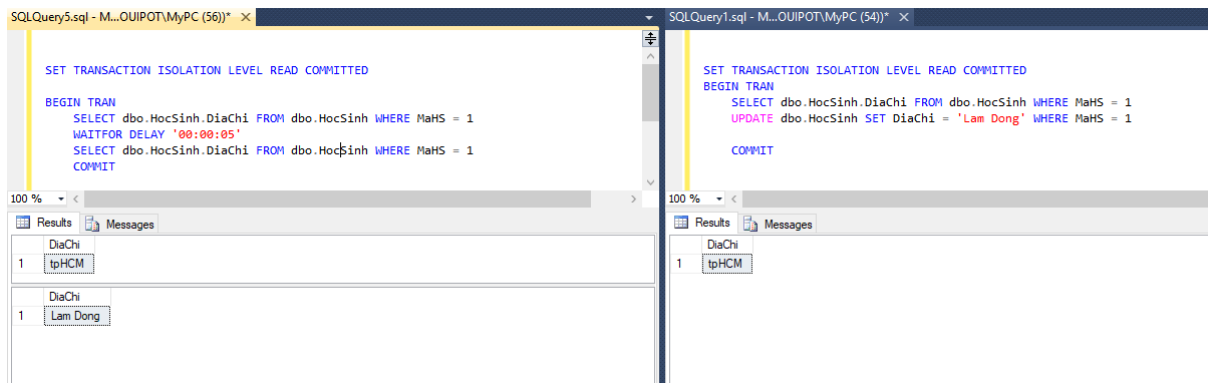
Results Messages

	MaGV	TenGV	NgaySinh	DiaChi	SDT	CMND	TrangThai
1	2	Phạm Quốc Cường	2018-12-20 00:00:00	Bình Dương	02165	5465132	1
2	3	Nguyễn Phạm Hương	2018-12-17 00:00:00	HCM	0231322	01123	1
3	4	Dương Văn Nam	2018-12-17 00:00:00	Đồng Nai	01123	032123	1

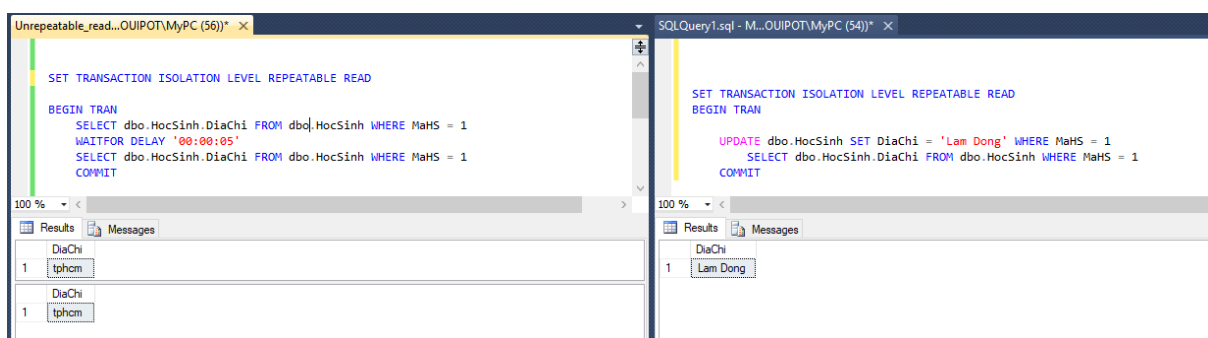
### 6.2.3. Unrepeatable read

#### ❖ Trường hợp 1

- Khi thay đổi địa chỉ của một học sinh. Ban đầu T1 đọc thông tin học sinh, sau đó delay. T2 vào thay đổi thông tin học sinh, sau đó commit lên CSDL. T1 trở lại không thể đọc lại dữ liệu nhưng không thể nhìn thấy dữ liệu ban đầu



- Hướng giải quyết: Dùng mức cô lập Repeatable Read. Khi chạy dòng lệnh select sẽ tạo Share lock và giữ Share lock này đến hết giao tác. Khi T2 vào update thông tin. T2 phải đợi T1 thực hiện xong mới update. → Dữ liệu đúng.



- ❖ Trường hợp 2: Khi T1 đọc dữ liệu Tên người giám hộ của học sinh có mã học sinh là 1 thì tên người giám hộ là “Không” (nhưng chưa kịp commit) thì có giao tác T2 thực hiện cập nhật Tên người giám hộ thành “Không cha”
  - Lúc này T1 đọc lại Tên người giám hộ không còn là “Không” nữa mà là “Không cha”

<pre> begin tran select TenNguoiGiamHo from HocSinh where MaHS=1 waitfor delay '00:00:10' select TenNguoiGiamHo from HocSinh where MaHS=1 commit </pre>	
90 %	
Results	Messages
	TenNguoiGiamHo
1	Không
	TenNguoiGiamHo
1	Không Cha

<pre> begin tran select TenNguoiGiamHo from HocSinh where MaHS=1 update HocSinh set TenNguoiGiamHo=(TenNguoiGiamHo+N' Cha') where MaHS=1 select TenNguoiGiamHo from HocSinh where MaHS=1 commit </pre>	
90 %	
Results	Messages
	TenNguoiGiamHo
1	Không
	TenNguoiGiamHo
1	Không Cha

- Nguyên nhân: Do T1 đang đọc mà T2 thực hiện update
- Giải quyết: sử dụng chế độ Repeatable Read, giữ shared lock đến hết giao tác => T1 thực hiện xong mới đến T2

```

set transaction isolation level repeatable read
begin tran
select TenNguoiGiamHo from HocSinh where MaHS=1
waitfor delay '00:00:10'
select TenNguoiGiamHo from HocSinh where MaHS=1
commit

```

90 %

Results Messages

	TenNguoiGiamHo
1	Không

	TenNguoiGiamHo
1	Không

```

begin tran
select TenNguoiGiamHo from HocSinh
where MaHS=1
update HocSinh set TenNguoiGiamHo=(TenNguoiGiamHo+N' Cha')
where MaHS=1
select TenNguoiGiamHo from HocSinh where MaHS=1
commit

```

90 %

Results Messages

	TenNguoiGiamHo
1	Không

	TenNguoiGiamHo
1	Không Cha

- ❖ Trường hợp 3: Cô hiệu trưởng cần thông tin số điện thoại của một giáo viên. Trong lúc xem, một nhân viên tiến hành cập nhật lại SĐT của giáo viên này.



⇒ Cô hiệu trường gọi không được.

```
--Unrepeatable data
--T1
BEGIN TRAN
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
WAITFOR DELAY '00:00:10'
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT
```

SDT
11111

SDT
012345

```
--Unrepeatable data
--T2
BEGIN TRAN
UPDATE dbo.GiaoVien SET SDT = N'012345' WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT
```

(1 row(s) affected)

- Nguyên nhân: Do T1 đang đọc giá trị , T2 vào ghi dữ liệu mới.
- Giải pháp: Chuyển mức cô lập của T1 thành repeatable read. Khi đó T2 phải đợi T1 kết thúc mới update được.

```
--Unrepeatable data
--T1
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
WAITFOR DELAY '00:00:05'
SELECT SDT FROM dbo.GiaoVien WHERE TenGV=N'Nguyễn Phạm Hương'
COMMIT
```

161 %

Results Messages

	SDT
1	012345

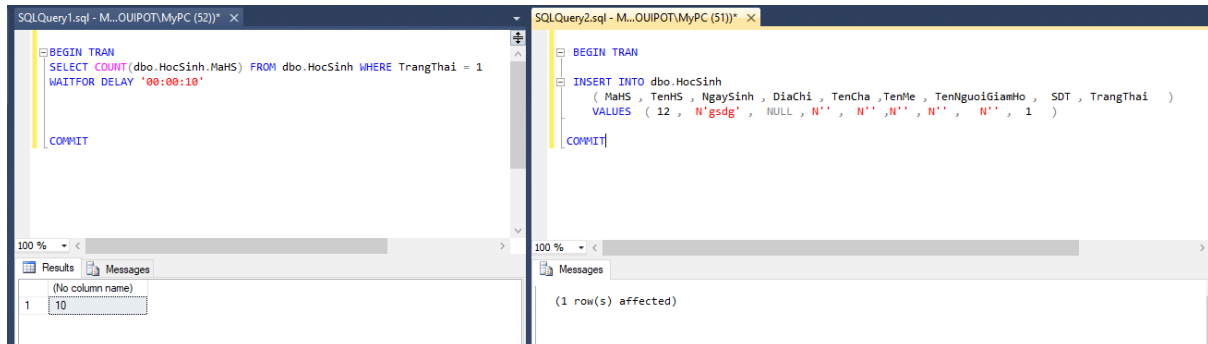
  

	SDT
1	012345

## 6.2.4. Phantom

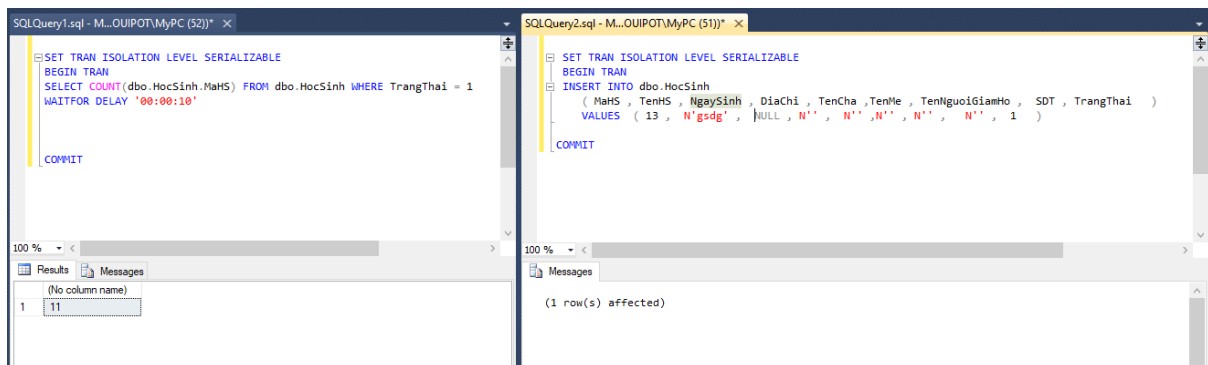
### ❖ Trường hợp 1:

Trong khi T1 đang xem tổng số học sinh, T2 thực hiện chèn thêm 1 học sinh. Điều này làm thông tin khi T1 nhìn thấy bị sai → Xảy ra hiện tượng Phantom

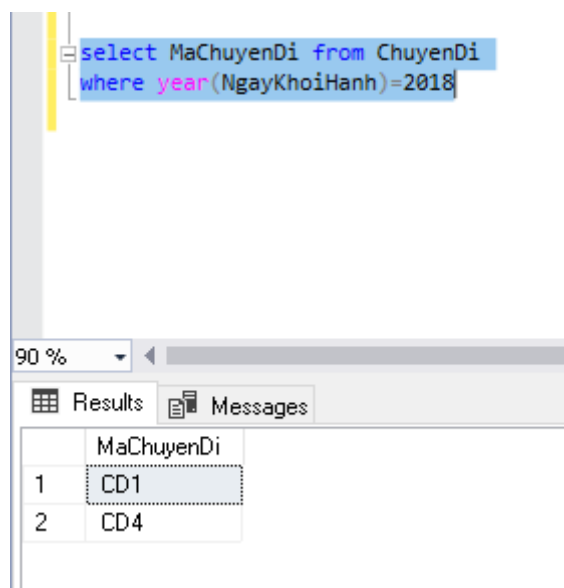


Tại đây. Số học sinh phải là 11. Tuy nhiên tại T1 chỉ nhìn thấy là 10 học sinh

- Hướng giải quyết : Dùng mức cô lập Serializable. T2 phải chờ đến khi T1 thực hiện xong rồi mới thực hiện.



### ❖ Trường hợp 2: T1 đang thực hiện đếm số chuyến đi trong năm 2018 thì T2 thực hiện insert thêm 1 chuyến đi trong năm 2018 làm T1 sai kết quả thao tác của T1



```

begin tran
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
waitfor delay '00:00:05'
commit

select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018

```

90 %

Results Messages

	(No column name)
1	2

	(No column name)
1	3

```

begin tran
insert into ChuyenDi(MaChuyenDi,NgayKhoiHanh)
values ('CD10','1/1/2018')
commit

```

90 %

Messages

(1 row affected)

- **Nguyên nhân:** T1 đang thực hiện đọc dữ liệu thì T2 cập nhật thêm dữ liệu làm việc đọc của T1 bị sai
- **Giải quyết:** Chuyển mức cô lập của T1 về serializable. T1 không cho phép T2 Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock)  
⇒ T1 làm xong rồi mới tới T2 thao tác

```

set transaction isolation level serializable
begin tran
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
waitfor delay '00:00:05'
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018
commit

```

90 %

Results Messages

	(No column name)
1	2

	(No column name)
1	2

```

begin tran
insert into ChuyenDi(MaChuyenDi,NgayKhoiHanh)
values ('CD10','1/1/2018')
commit
select count(MaChuyenDi) from ChuyenDi
where year(NgayKhoiHanh)=2018

```

90 %

Results Messages

	(No column name)
1	3

- ❖ Trường hợp 3: Tình huống: Trong lúc đang tính tổng số tiền đã thanh toán cho hợp đồng HD1 thì có 1 nhân viên khác thêm thông tin thanh toán cho hợp đồng này  
⇒ Kết quả tính toán bị sai.

```
--phantom
--T1

BEGIN TRAN
    SELECT SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=N'HD1'
    WAITFOR DELAY '00:00:05'|
COMMIT
SELECT SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=N'HD1'
```

161 %

Results Messages

(No column name)
1 200000

(No column name)
1 210000

```
--phantom
--T2

BEGIN TRAN
    INSERT INTO dbo.ThongTinThanhToan
        ( MaHopDong ,
          LanThanhToan ,
          NgayThanhToan ,
          SoTien
        )
    VALUES ( N'HD1' , -- MaHopDong - nvarchar(6)
             4 , -- LanThanhToan - int
             GETDATE() , -- NgayThanhToan - date
             10000 -- SoTien - float
            )
COMMIT
```

- **Nguyên nhân:** Trong quá trình T1 đang thực hiện tính toán thì T2 thêm 1 dòng dữ liệu vào bảng thông tin thanh toán.
- **Giải pháp:** Chuyển mức cô lập về Serializable. T2 phải chờ đến khi T1 kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này. Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock)

```
--phantom
--T1
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
BEGIN TRAN
    SELECT SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=N'HD1'
    WAITFOR DELAY '00:00:05'
    SELECT SUM(SoTien) FROM dbo.ThongTinThanhToan WHERE MaHopDong=N'HD1'
COMMIT
```

146 %

Results Messages

	(No column name)
1	210000

	(No column name)
1	210000

### 6.3. Deadlock

- Mô tả tình huống: khi T1 thực hiện việc cập nhật thông tin của học sinh A, sau đó delay 10s. T2 vào thực hiện cập nhật giáo viên 1 sau đó delay. T1 trở lại cập nhật giáo viên 1. T2 sau delay trở lại cập nhật học sinh A. Điều này khiến cho T1 chờ tài nguyên của T2 (giáo viên). T2 chờ tài nguyên của T1( học sinh). → xảy ra deadlock.

SQLQuery1.sql - M...OUIPOT\MyPC (53)\*

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRAN
UPDATE dbo.HocSinh SET DiaChi = 'TPHCM' WHERE MaHS = 1
WAITFOR DELAY '00:00:10'
UPDATE dbo.GiaoVien SET DiaChi = 'Thai Binh' WHERE MaGV = 1
COMMIT
SELECT * FROM dbo.HocSinh WHERE MaHS = 1
SELECT * FROM dbo.GiaoVien WHERE MaGV = 1
```

MaHS	TenHS	NgaySinh	DiaChi	TenCha	TenMe	TenNguoiGiamHo	SDT
1	Nguyễn Văn A	2013-02-05 00:00:00	TPHCM	Nguyễn Tiến A	Nguyễn Thảo A	Không	0132

MaGV	TenGV	NgaySinh	DiaChi	SDT	CMND	TrangThai
1	Nguyễn Văn Loc	2018-03-02 00:00:00	Thai Binh	234567890	456789	1

SQLQuery1.sql - M...OUIPOT\MyPC (52)\*

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
BEGIN TRAN
UPDATE dbo.GiaoVien SET DiaChi = 'Thai Binh' WHERE MaGV = 1
WAITFOR DELAY '00:00:10'
UPDATE dbo.HocSinh SET DiaChi = 'TPHCM' WHERE MaHS = 1
COMMIT
```

(1 row(s) affected)  
Msg 1205, Level 13, State 51, Line 7  
Transaction (Process ID 52) was deadlocked on lock resources with another process and h

SQLQuery1.sql - M...OUIPOT\MyPC (53)\*

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
UPDATE dbo.HocSinh SET DiaChi = 'TPHCM' WHERE MaHS = 1
WAITFOR DELAY '00:00:10'
UPDATE dbo.GiaoVien SET DiaChi = 'Thai Binh' WHERE MaGV = 1
COMMIT
SELECT * FROM dbo.HocSinh WHERE MaHS = 1
SELECT * FROM dbo.GiaoVien WHERE MaGV = 1
```

MaHS	TenHS	NgaySinh	DiaChi	TenCha	TenMe	TenNguoiGiamHo	SDT
1	Nguyễn Văn A	2013-02-05 00:00:00	TPHCM	Nguyễn Tiến A	Nguyễn Thảo A	Không	

MaGV	TenGV	NgaySinh	DiaChi	SDT	CMND	TrangThai
1	Nguyễn Văn Loc	2018-03-02 00:00:00	Thai Binh	234567890	456789	1

SQLQuery1.sql - M...OUIPOT\MyPC (52)\*

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
BEGIN TRAN
UPDATE dbo.GiaoVien SET DiaChi = 'Thai Binh' WHERE MaGV = 1
WAITFOR DELAY '00:00:10'
UPDATE dbo.HocSinh SET DiaChi = 'TPHCM' WHERE MaHS = 1
COMMIT
```

(1 row(s) affected)  
Msg 1205, Level 13, State 51, Line 7  
Transaction (Process ID 52) was deadlocked on lock resources with another process and h

- Vấn đề: T1 và T2 cùng xem địa điểm DD5 và thấy địa chỉ bị sai nên cả 2 đồng thời sửa. Mức cô lập là Repeatable read để chắc chắn giữ shared lock đến hết giao tác tránh xung đột. Nhưng lại xảy ra deadlock.



```

begin tran
set transaction isolation level repeatable read
select DiaChi from DiaDiem where MaDiaDiem='DD5'
waitfor delay '00:00:05'
update DiaDiem
set DiaChi = N'28 Võ Thị Sáu'
where MaDiaDiem='DD5'
select DiaChi from DiaDiem where MaDiaDiem='DD5'
commit

```

90 %

Results Messages

	DiaChi
1	28 Lý Tự Trọng

	DiaChi
1	28 Võ Thị Sáu

```

begin tran
set transaction isolation level repeatable read
select DiaChi from DiaDiem where MaDiaDiem='DD5'
update DiaDiem
set DiaChi = N'28 Trần Văn Ôn'
where MaDiaDiem='DD5'
select DiaChi from DiaDiem where MaDiaDiem='DD5'
commit

```

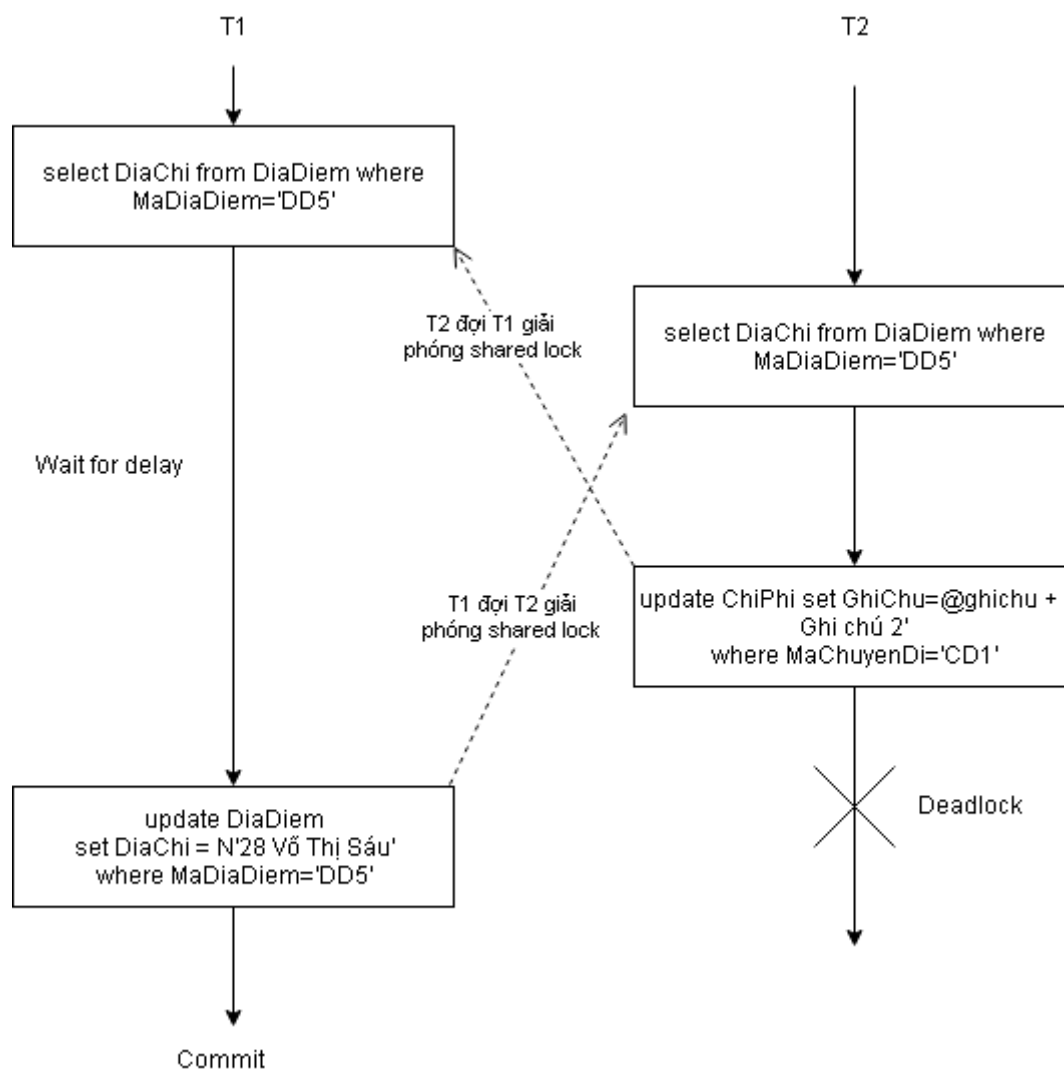
0 %

Results Messages

(1 row affected)

Msg 1205, Level 13, State 51, Line 4  
Transaction (Process ID 56) was deadlocked on lock resources with another transaction and has been chosen as the deadlock victim. ROLLBACK TRANSACTION

**Nguyên nhân:**



**Giải pháp:** Thêm Update lock vào lệnh select => T2 phải chờ T1 chạy xong mới được chạy.

```
begin tran
set transaction isolation level repeatable read
select DiaChi from DiaDiem with(updlock) where MaDiaDiem='DD5'
waitfor delay '00:00:05'
update DiaDiem
set DiaChi = N'28 Võ Thị Sáu'
where MaDiaDiem='DD5'
select DiaChi from DiaDiem where MaDiaDiem='DD5'
commit
```

	DiaChi
1	28 Lý Tự Trọng

	DiaChi
1	28 Võ Thị Sáu

```
begin tran
set transaction isolation level repeatable read
select DiaChi from DiaDiem with(updlock) where MaDiaDiem='DD5'
update DiaDiem
set DiaChi = N'28 Trần Văn Ôn'
where MaDiaDiem='DD5'
select DiaChi from DiaDiem where MaDiaDiem='DD5'
commit
```

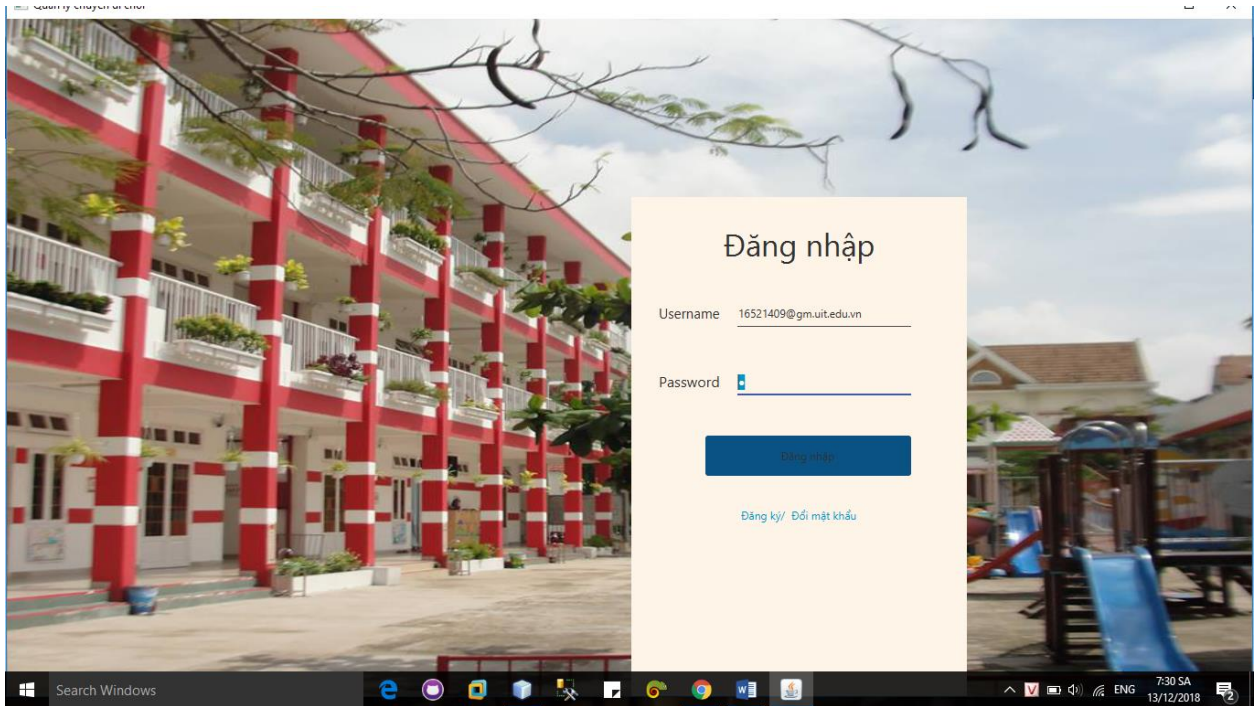
	DiaChi
1	28 Võ Thị Sáu

	DiaChi
1	28 Trần Văn Ôn

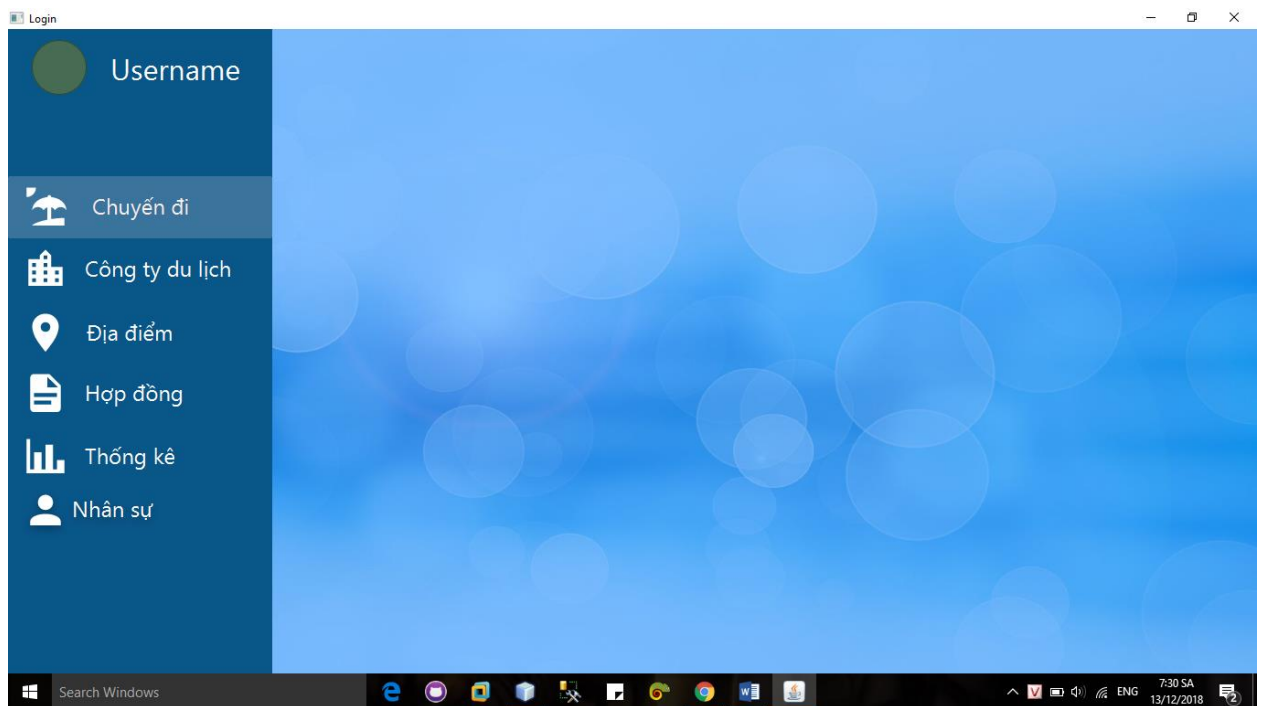
## 7. Thiết kế giao diện

Danh sách các màn hình

- Form giao diện



- Giao diện chính



- Chuyển đi

[illegible]

- Chi tiết chuyến đi

[illegible]

- Chi phí

Username

## Quản lý chi phí

### Chuyến đi: CD1

Vé cồng:	1000000.0	đồng	Nước uống:	4567.0	đồng
Tiền xe:	1000.0	đồng	Tiền ăn xe:	567.0	đồng
Tiền ăn trưa:	4567890.0	đồng	Linh tinh:	67.0	đồng
Phi hướng dẫn viên:	1.2345678E7	đồng	Tổng tiền:	1.7919768E7	đồng

Ghi chú: Ghi chú ở đây:

Lưu

Search Windows

7:54 SA 13/12/2018

- Tham gia

The screenshot shows a web application titled "Quản lý tham gia". On the left is a dark blue sidebar with a "Login" button and a list of menu items: "Chuyến đi", "Công ty du lịch", "Địa điểm", "Hợp đồng", "Thống kê", "Nhân sự", "Học Sinh", and "Giáo viên". The main content area has a header with "Username" and a dropdown menu for "Lớp" (Class) set to "Mầm 1". Below this are two tables. The first table, "Tham gia", lists students with columns for "Mã học sinh", "Tên học sinh", "Tham gia", and "Đóng tiền". The second table, "Tham gia", lists teachers with columns for "Mã giáo viên", "Tên giáo viên", and "Tham gia". A "Lưu" (Save) button is at the bottom right.

Mã học sinh	Tên học sinh	Tham gia	Đóng tiền
3	Nguyễn Bà C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Lê Sáng F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Vũ Văn I	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

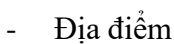
Mã giáo viên	Tên giáo viên	Tham gia
	Nguyễn Văn Lọc	<input checked="" type="checkbox"/>

- Công ty du lịch

The screenshot shows a web application titled "QUẢN LÝ CÔNG TY DU LỊCH". The sidebar is identical to the previous one. The main content area has a header with "Username" and a search bar with the placeholder "Nhập dữ liệu cần tìm kiếm:" and a "Tìm" (Search) button. Below the search bar are input fields for "Mã công ty:", "Tên công ty:", "Địa chỉ:", and "SDT:". There are "Mới" (New) and "Sửa" (Edit) buttons. To the right is a table with columns "Mã công ty", "Tên", "Địa chỉ", and "SDT".

Mã công ty	Tên	Địa chỉ	SDT
CT1	VietTravel	190 Pasteur, Quận 3, HCM, VietNam	19001839
CT2	Saigontourist	45, Le Thanh Ton, Ben Nghe, TP H...	19001808







## - Hợp đồng

Login

Username

Chuyển đi

Công ty du lịch

Địa điểm

Hợp đồng

Thống kê

Nhân sự

Học Sinh

Giáo viên

QUẢN LÝ HỢP ĐỒNG

Mã hợp đồng:

\* Thông tin hợp đồng

Mã hợp đồng	Mã chuyển đi	Công ty du lịch	Trị giá	Trang thái	Ngày ký
HD1	CD1	VietTravel	10000.0	Đã th...	2018-12-01

Mã hợp đồng:

Mã chuyển đi:

Công ty du lịch:

Ngày ký:

Trị giá:

\* Thông tin thanh toán

Mã hợp đồng	Lần thanh toán	Ngày thanh toán	Số tiền
No content in table			

Mã hợp đồng:

Lần thanh toán:

Ngày thanh toán:

Trị giá:

Tổng đã thanh toán:  Còn nợ:

Search Windows

8:00 SA

13/12/2018

Login

Username

Chuyển đi

Công ty du lịch

Địa điểm

Hợp đồng

Thống kê

Nhân sự

Học Sinh

Giáo viên

QUẢN LÝ HỢP ĐỒNG

Mã hợp đồng:

\* Thông tin hợp đồng

Mã hợp đồng	Mã chuyển đi	Công ty du lịch	Trị giá	Trang thái	Ngày ký
HD1	CD1	VietTravel	10000.0	Đã th...	2018-12-01

Mã hợp đồng: HD1

Mã chuyển đi: CD1

Công ty du lịch: VietTravel

Ngày ký: 01/12/2018

Trị giá: 10000.0

\* Thông tin thanh toán

Mã hợp đồng	Lần thanh toán	Ngày thanh toán	Số tiền
HD1	1	2018-12-20	5000.0
HD1	2	2018-12-20	5000.0

Mã hợp đồng: HD1

Lần thanh toán:

Ngày thanh toán:

Trị giá:

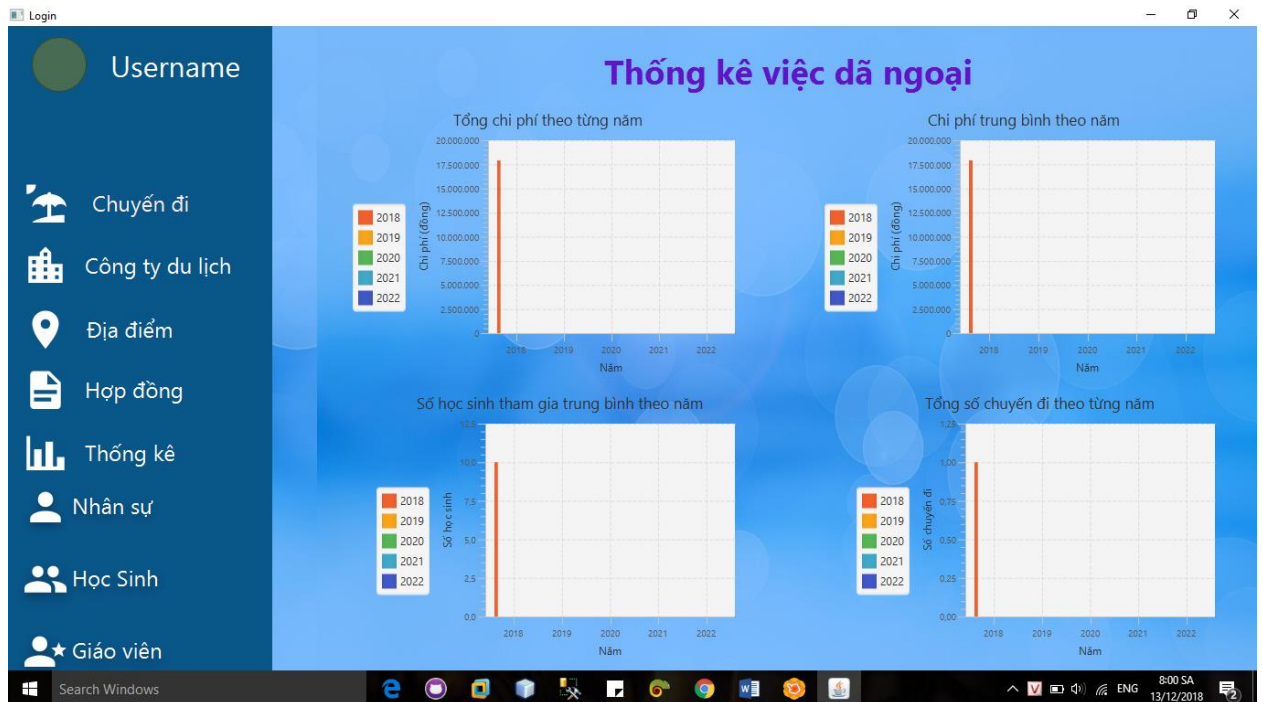
Tổng đã thanh toán: 10000.0 Còn nợ: 0.0

Search Windows

8:00 SA

13/12/2018

- Thống kê



- Học sinh

**QUẢN LÝ HỌC SINH**

Lớp học:  Năm Học:

Tên học sinh:

Mã học sinh:

Tên học sinh:

Ngày sinh:

Địa chỉ:

Tên cha:

Tên Mẹ:

Tên người giám hộ:

Số điện thoại:

Mã học sinh	Tên	Ngày sinh	Địa chỉ	Tên Cha	Tên Mẹ	SDT
5	Nguyễn Quà E	2014-08-05	Long An	Nguyễn Quà...	Nguyễn Ngọc E	01475896235
8	Đào Bá H	2014-12-25	An Giang	Đào Văn H	Trần Thị Văn H	0586589825

- Giáo viên

[illegible]

## 8. Kiểm tra phần mềm

### 8.1. Mục đích của việc kiểm tra phần mềm

Báo cáo kiểm tra phần mềm Quản lý chuyến đi dã ngoại của trường mẫu giáo Thiên Thần cung cấp các kết quả kiểm thử một cách khách quan trên bộ dữ liệu mẫu được lấy ngẫu nhiên (Không theo sắp xếp). Các kết quả nhận được phản ánh mức độ hoàn thành phần mềm và tính đúng đắn của hệ thống, đảm bảo hệ thống hoạt động một cách an toàn và hiệu quả khi được ứng dụng trong thực tế.

### 8.2. Các yêu cầu kiểm tra

Danh sách sau liệt kê các đối tượng được tập trung kiểm tra trong các Usecase, các yêu cầu chức năng và các yêu cầu phi chức năng đối với hệ thống.

- Đăng nhập hệ thống
- Quản lý học sinh
- Quản lý giáo viên
- Quản lý địa điểm
- Quản lý chuyến đi
- Quản lý chi phí
- Quản lý chi tiết chuyến đi
- Quản lý tham gia
- Thống kê chi phí, học sinh, chuyến đi
- Quản lý hợp đồng

### 8.3. Các chiến lược kiểm tra

#### 8.3.1. Kiểm tra tích hợp dữ liệu và cơ sở dữ liệu

##### 8.3.1.1. Mục tiêu

Đảm bảo các phương thức truy cập cơ sở dữ liệu và các chức năng xử lý hoạt động đúng và dòng dữ liệu không bị ngắt giữa chừng.

##### 8.3.1.2. Kỹ thuật

Mọi phương thức truy cập cơ sở dữ liệu và chức năng xử lý được kiểm tra trên các bộ dữ liệu test cả đúng và cả không đúng.

##### 8.3.1.3. Kiểm tra phần mềm quản lý chuyến đi dã ngoại trường mẫu giáo Thiên Thần

Cơ sở dữ liệu của siêu thị được lần lượt kiểm tra truy cập và tiến hành xử lý theo các đối tượng đã được nêu ở trên. Mọi dữ liệu và thông tin cần thiết được truy cập và lấy ra trong thời gian hợp lý. Việc xử lý tiến hành đúng đắn. Các bộ phận dữ liệu đúng được đưa vào kiểm tra, tiếp theo là các bộ dữ liệu không chính xác cũng được đưa vào kiểm tra. Đối với dữ liệu đúng đắn, hệ thống thực hiện các chức năng theo như yêu cầu đặt ra, đối với các bộ dữ liệu không đúngn hệ thống yêu cầu nhập lại dữ liệu và không tiến hành các xử lý. Sau khi kết thúc hệ thống, các dữ liệu mang tính Persistency tiếp tục tồn tại một cách độc lập, đảm bảo tính bền vững của dữ liệu.

#### **8.3.1.4. Tiêu chuẩn hoàn chỉnh**

Tất cả các phương thức truy cập cơ sở dữ liệu và các hàm xử lý thực hiện đúng theo thiết kế và không bị ngắt giữa chừng trong khi truy cập cơ sở dữ liệu.

### **8.3.2. Kiểm tra các chức năng hệ thống**

#### **8.3.2.1. Mục tiêu**

Đảm bảo tính đúng đắn của các chức năng yêu cầu của hệ thống (đã được liệt kê), bao gồm chiều của các luồng thông tin, dữ liệu nhập, xử lý và trả kết quả.

#### **8.3.2.2. Kỹ thuật**

Chiến lược kiểm tra các chức năng sử dụng kỹ thuật kiểm tra hộp đen, quan tâm đến ứng xử của hệ thống và giao tác với người sử dụng hệ thống, thẩm định ứng dụng và các xử lý bên trong của nó khi tương tác với người sử dụng thông qua giao diện đồ họa, cuối cùng là phân tích kiểm chứng các kết quả xuất với các kết quả được tính bằng tay trước đó.

Đối với mọi chức năng của hệ thống cần sử dụng các bộ dữ liệu khác nhau cả đúng và cả sai để kiểm tra và ghi nhận ứng xử của hệ thống.

#### **8.3.2.3. Kiểm tra phần mềm quản lý chuyển đi đã ngoại trường mẫu giáo Thiên Thần**

Các chức năng được liệt kê trong "các yêu cầu kiểm tra" được lượng kiểm tra. Đối với các bộ dữ liệu đúng hệ thống cho kết quả tính toán chính xác (tính tổng chi phí, thống kê chi phí, học sinh). Về chức năng lưu trữ cơ sở dữ liệu được thêm/sửa/xóa hoàn chỉnh, không xảy ra tình trạng mất mát dữ liệu và hiển thị không đúng. Về chức năng tra cứu thống kê hệ thống cho thấy được đầy đủ các thông tin cần thiết về cho

kết quả nhớ được tính bằng tay trước. Với các bộ dữ liệu không hợp lệ hệ thống yêu cầu nhập lại các thông tin không hợp lệ các ràng buộc dịch vụ cũng được xem xét về hệ thống đã có thể phản ứng tốt trong các tình huống yêu cầu tự động nhận biết các nhiệm vụ chức năng như khi chuyển đi chưa có chi phí, hệ thống sẽ tự tạo một chi phí rỗng cho chuyển đi đó.

#### **8.3.2.4. Tiêu chuẩn hoàn chỉnh**

Tất cả các đối tượng kiểm tra đều thực hiện tốt chức năng của mình, và các sai sót được chỉ ra rõ ràng.

### **8.3.3. Kiểm tra giao diện người dùng**

#### **8.3.3.1. Mục tiêu**

Thẩm định các tương tác của người sử dụng với phần mềm. Mục đích chính của việc kiểm tra giao diện người dùng là đảm bảo hệ thống cung cấp các giao diện người dùng đúng và các truy cập chính xác cũng như chiều của các luồng sự kiện của các đối tượng cần kiểm tra.

Kiểm tra tính đúng đắn của các chức năng của phần mềm thông qua việc giám sát các chức năng yêu cầu hệ thống có, bao gồm từng cửa sổ tương tác, các trường dữ liệu và các phương thức truy cập hệ thống như phím Tab, sự kiện di chuyển chuột, các phím tắt... Kiểm tra tất cả các đối tượng cửa sổ cùng với các đặc tính của nó.

#### **8.3.3.2. Kỹ thuật**

Kiểm tra tạo lập và sửa đổi từng cửa sổ để thẩm định tính đúng đắn của chiều xử lý và các trạng thái của đối tượng cho mỗi đối tượng và cửa sổ ứng dụng.

#### **8.3.3.3. Kiểm tra phần mềm quản lý chuyển đi đã ngoại trường mẫu giáo Thiên Thần**

Tất cả các cửa sổ trong phần mềm lần được lần lượt kiểm tra, các chức năng cơ bản như định hướng xử lý, phản hồi phím nóng, các sự kiện chuột trên cửa sổ ứng dụng đều khá hoàn chỉnh và không làm ảnh hưởng đến kết quả xử lý của hệ thống. Trong mỗi cửa sổ kiểm tra các Focus mặc định và các Focus khi di chuyển con trỏ bàn phím Tab. Các phím tắt không bị đụng nhau trong mỗi cửa sổ. Giao diện thân thiện và được bố trí các thành phần hợp lý chuyển đổi giữa các giao diện được thực hiện tốt.

#### **8.3.3.4. Tiêu chuẩn hoàn chỉnh**

Các cửa sổ được thẩm định thành công trong việc duy trì sự tương thích thao tác của người sử dụng và cung cấp đầy đủ các chức năng theo như yêu cầu.

### **8.3.4. Kiểm tra bảo mật và điều khiển truy cập**

#### **8.3.4.1. Mục đích**

Kiểm tra:

- Bảo mật ở mức ứng dụng: bao gồm các truy cập dữ liệu và các chức năng hệ thống.
- Bảo mật ở mức hệ thống: bao gồm đăng nhập vào hệ thống hoặc điều khiển truy cập.

Đảm bảo hệ thống phân quyền tốt, mỗi người sử dụng với quyền đăng nhập khác nhau sẽ có các quyền truy cập hệ thống và cơ sở dữ liệu khác nhau. Đảm bảo tính toàn vẹn và đồng bộ dữ liệu của phần mềm. Các thông tin nhạy cảm và bí mật cần có cơ chế truy cập không hợp phép.

#### **8.3.4.2. Kỹ thuật**

Liệt kê danh sách các người sử dụng có tương tác với hệ thống (actor) ứng với các chức năng và dữ liệu được phép truy cập khác nhau.

Lần lượt kiểm tra cho từng loại người truy cập hệ thống bằng cách thực hiện các giao tác đặc biệt ứng với mỗi đối tượng truy cập hệ thống.

Lập lại việc kiểm tra đối với mỗi đối tượng truy cập hệ thống nhưng với các quyền truy cập không được phép. Ghi nhận các phản ứng của hệ thống.

#### **8.3.4.3. Kiểm tra phần mềm Quản lý chuyển đi đã ngoại trường mẫu giáo Thiên Thần**

Hệ thống được bảo vệ thông qua chức năng đăng nhập hệ thống. Có hai loại đối tượng có tương tác với hệ thống: hiệu trưởng và nhân viên ứng với các quyền truy cập hệ thống và dữ liệu như trong bảng sau:



Chức năng	Dữ liệu	Hiệu trưởng	Nhân viên
Đăng nhập		x	x
Quản lý học sinh	Thông tin về học sinh	x	x
Quản lý giáo viên	Thông tin về giáo viên	x	
Quản lý địa điểm	Thông tin về địa điểm	x	x
Quản lý chuyến đi	Thông tin về chuyến đi	x	x
Quản lý hợp đồng	Thông tin về hợp đồng	x	
Quản lý tham gia	Thông tin về việc tham gia của học sinh và giáo viên	x	x
Quản lý chi tiết chuyến đi	Thông tin về chi tiết chuyến đi	x	x
Thống kê chuyến đi	Thông tin về chi phí, học sinh, số chuyến đi	x	

Với quyền ứng phần mềm chỉ cho phép truy cập đến các chức năng cho phép, người đăng nhập được thực hiện hoàn chỉnh và không xảy ra lỗi đăng nhập. Khi có thấy có sai sót trong công tác đăng nhập (sai về email hoặc sai về mật khẩu) hệ thống sẽ ngăn cản việc đăng nhập và phản hồi với người sử dụng.



## 9. Tổng kết – đánh giá

Phần mềm quản lý chuyến đi dã ngoại của trẻ mầm non giúp người dùng dễ dàng quản lý thông tin về các chuyến đi chơi của trẻ. Qua đó có những ưu nhược điểm sau:

- Ưu điểm:
  - Phần mềm đáp ứng được hầu hết các yêu cầu của khách hàng như quản lý về nhân sự, quản lý về chuyến đi của trẻ mầm non.
  - Giúp quý Thầy cô dễ dàng sử dụng, tiện lợi trong việc tra cứu, thống kê, quản lý sổ sách.
  - Lưu và tải dữ liệu nhanh chóng
  - Dễ sử dụng, không đòi hỏi kiến thức về tin học
  - Dễ dàng sửa và bảo trì hệ thống
- Tuy nhiên vẫn còn một số hạn chế sau:
  - Chưa thể in báo cáo ra file pdf cho người sử dụng
  - Các transaction chưa thực tế
  - Chưa làm phần hướng dẫn sử dụng
  - Form chưa đẹp và bắt mắt
  - Phần mềm offline
- Hướng phát triển:
  - Giải quyết xung đột thực tế hơn.
  - Tạo thêm nhiều tính năng gợi ý cho người sử dụng.
  - Tạo tính năng tự động gửi email cho phụ huynh học sinh.
  - Phát triển thành phần mềm online
  - Thiết kế giao diện đẹp hơn