****

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| Đỗ Đức Minh Triết | 23521650 |
| Nguyễn Quốc Việt | 23521782 |
| Nguyễn Thanh Tùng | 23521744 |

GVHD: **Phan Trung Phát**

Lớp: **NT106.P13.1**

6

**Contest**

**ĐÁNH GIÁ KHÁC (\*):**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình (1) | 2 ngày |
| Link Video thực hiện (2)  *(nếu có)* | [Lab6](https://drive.google.com/file/d/18ZOl36wVsd78daopm2JOpLUw6qzhYw-Z/view?usp=sharing) |
| Ý kiến (3) *(nếu có)*  + Khó khăn  + Đề xuất … |  |
| Điểm tự đánh giá (4) | 8.5/10 |
| (\*): phần (1) và (4) bắt buộc thực hiện. | |

# Câu hỏi 4

## 1. Minh chứng:

## A screenshot of a computer Description automatically generated

Hình . Hình ảnh giao diện bên phía client

A screen shot of a computer program

Description automatically generated

Hình . Class MovieTicket

A black background with white text

Description automatically generated

Hình . Khai báo List luồng, hàm cancel, IP, Socket

A computer screen shot of a program

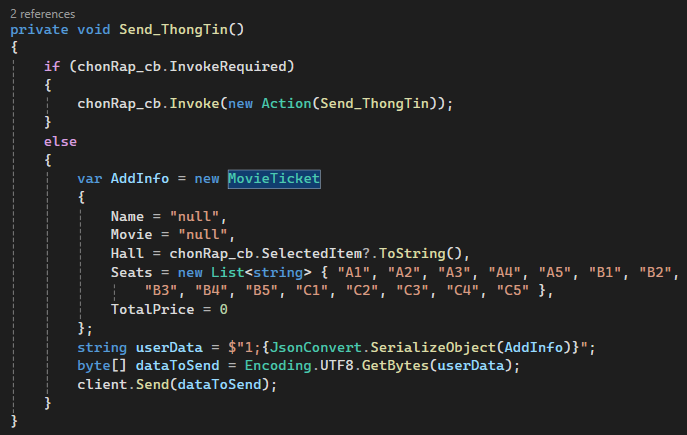
Description automatically generated

Hình . Hàm Connect đến server

A black screen with white text

Description automatically generated

Hình . Hàm đóng kết nối



Hình . Hàm Gửi thông tin

A computer screen shot of a program code

Description automatically generated

Hình . Hàm gửi thông tin khi đặt vé

A computer screen shot of a program code

Description automatically generated

Hình . Hàm Deserialize

A screen shot of a computer program

Description automatically generated

A computer screen with many colorful text

Description automatically generated

Hình . Hàm Receive thông tin từ server

A screen shot of a computer program

Description automatically generated

Hình . Initilize các giá trị, hàm khi bắt đầu chương trình

A screen shot of a computer program

Description automatically generated

Hình . Class MovieStatistics

A screen shot of a computer program

Description automatically generated

Hình . Class ThongTin

A computer screen shot of a program code

Description automatically generated

Hình . Class Hall, List my\_list với type là ThongTin, Dictionary my\_hall

A screen shot of a computer program

Description automatically generated

Hình . Function Lưu thông tin

A computer screen shot of a program code

Description automatically generated

Hình . Nội dung button Lưu và hàm Deserialize Thông Tin

A computer screen shot of a black background

Description automatically generated

Hình . Hàm kiểm tra dấu phẩy ở input

A screen shot of a computer program

Description automatically generated

A computer screen with colorful text

Description automatically generated

Hình . Nội dung button Add

A screen shot of a computer code

Description automatically generated

Hình . Button Lấy thông tin

A computer screen shot of a program code

Description automatically generated

Hình . Activity trong comboBox Chọn phim

A screen shot of a computer program

Description automatically generated

Hình . Activity trong comboBox Chọn rạp

A screen shot of a computer program

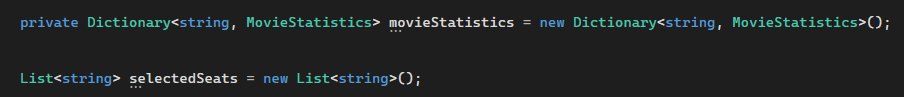
Description automatically generated

Hình . Activity trong Event Chọn ghế checklistBox

A computer screen shot of a program code

Description automatically generated

Hình . Hàm kiểm tra ghế đã được đặt hay chưa



Hình . Dictionary MovieStatistics và List selectedSeats

A screen shot of a computer program

Description automatically generated

A computer screen with many colorful text

Description automatically generated

Hình . Activity trong button Đặt vé

A computer screen shot of a program code

Description automatically generated

Hình . Nội dung button XuatThongTin

A screen shot of a computer program

Description automatically generated

Hình . Đóng kết nối khi form đóng

A computer screen shot of a program

Description automatically generated

Hình . Kiểm tra Checked Seats khi chọn ghế

A screenshot of a computer

Description automatically generated

Hình . Giao diện server

A screen shot of a computer program

Description automatically generated

Hình . Khai báo các kết nối cần thiết, Initialize khi chương trình bắt đầu

A computer screen shot of a program

Description automatically generated

Hình . Hàm Khởi động kết nối cho server

A computer screen shot of a program code

Description automatically generated A computer screen shot of a program code

Description automatically generated

Hình . Nội dung hàm Receive bên server

A screen shot of a computer program

Description automatically generated

Hình . Hàm kiểm tra Rạp phim đã có trong Database hay chưa

A screen shot of a computer program

Description automatically generated

Hình . Hàm Insert ghế của rạp

A screen shot of a computer code

Description automatically generated

Hình . Hàm Update trạng thái ghế sau khi mua

A computer screen with colorful text

Description automatically generated

Hình . Hàm Deserialize và đóng kết nối bên server khi đóng form

A screenshot of a computer

Description automatically generated

Hình . Form Đăng nhập của SuperUser

A screenshot of a computer program

Description automatically generated

Hình . Nội dung của form Đăng nhập

A screenshot of a computer

Description automatically generated

Hình . Form Điều hướng

A screen shot of a computer program

Description automatically generated

Hình . Nội dung form Điều hướng

A screenshot of a computer

Description automatically generated

Hình . Giao diện của SuperUser

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer program

Description automatically generated

Hình . Nội dung form SuperUser

## 2. Giải thích:

Class **MovieTicket**:

- **Nội dung**: Kiểu dữ liệu này sẽ đóng vai trò thao tác với **Server.**

**- Trình bày:** Khai báo các biến Tên người dùng, tên phim, rạp đã chọn, số ghế đã chọn, tổng giá tiền, **IsInitialInfo** (kiểm tra xem **dữ liệu** được gửi đi có phải đã dữ liệu **khởi tạo** thông tin hay không), **IsOccupied** (Check xem dữ liệu ghế được gửi đi là khai báo **nhằm lưu** vào cơ sở dữ liệu hay **cập nhật** dữ liệu trên cơ sở dữ liệu).

List **Thread**:

- **Nội dung**: Theo dõi các luồng hoạt động => nhằm **quản lý** luồng khi đóng form, tránh luồng chạy ngầm.

Lớp **CancellationTokenSource**:

- **Nội dung**: Cơ chế báo hủy trong các hoạt động đa luồng hoặc không đồng bộ.

**IPEndPoint, Socket**:

- **Nội dung**: Khởi tạo IP, socket để kết nối đến server.

Hàm **ConnectToServer()**:

- **Nội dung**: Khai báo địa chỉ IP đích, client, kết nối đến Server.

- **Trình bày**:

- Khai báo địa chỉ đích của server **127.0.0.1 – địa chỉ LoopBack**, có thể đổi sang địa chỉ khác ví dụ **192.168.1.15** nếu server ở máy tính khác.

- Thread **listen** = new Thread:

* Khởi tạo luồng, chạy độc lập so với luồng chính.
* **Lamda Expression**: Định nghĩa anonymous method và truyền token **cancellationTokenSource.Token** làm tham số.
* Phương thức **Receive:** Xử lý dữ liệu đến từ server, CancellationToken cho phép dừng **Receive** khi cần thiết.
* **Threads.Add(listen)**: theo dõi tất cả các luồng được tạo.

Hàm **CloseConnection()**:  
- **Nội dung**: đóng kết nối phía client.

Hàm **Send\_ThongTin()**:

- **Nội dung**: Gửi nội dung khởi tạo nhằm để **server** cập nhật dữ liệu lên **database**.

- **Trình bày**: Sử dụng class **MovieTicket** để dễ dàng thống nhất dữ liệu gửi đi, Serialize dữ liệu bằng **JSON** tránh lỗi Input format ghi dùng **BinaryFormatter.**

Ở đây, **IsInitialInfo** được set bằng **true** và **IsOccupied** được set bằng **false** vì đây là hàm gửi nhằm khởi tạo thông tin trên cơ sở dữ liệu.

* Trước khi gửi đi, thông tin được thêm **Header = 1,** để **phân biệt** với các thông tin khác được gửi đến **server**.

Hàm **Send\_datVe()**:

- **Nội dung**: Update dữ liệu sau khi người dùng click **datVe button**.

- **Trình bày**: Truyền các đối tượng được người dùng chọn như tên phim, rạp phim, ghế được chọn sau đó gửi đến **server** để cập nhật thông tin trên **database.** Ở đây, **IsInitialInfo** được set bằng **false** và **IsOccupied** được set bằng **true** vì đây là hàm gửi nhằm cập nhật thông tin không phải khởi tạo thông tin.

* Trước khi gửi đi, thông tin được thêm **Header = 2,** để **phân biệt** với các thông tin khác được gửi đến **server**.

Hàm **Deserialize()**:

- **Nội dung**: **Deserialize** thông tin được truyền đến từ server (dạng **JSON**).

Hàm **Receive()**:

- **Nội dung**: Lắng nghe nội dung truyền đến từ server.

- **Trình bày**: Kiểm tra thông tin được nhận có phải là **Cancel Request** từ server hay không (**!token.IsCancellationRequested**) nếu không phải thì **Deserialize** dữ liệu. **Catch** các lỗi do **socket** hoặc lỗi khác xảy ra nếu có:

* **Text.StartsWith(“DONGMO”):** Kiểm tra **thông tin** nhận được có thuộc lệnh **Đóng/Mở** form, nếu thuộc lệnh **Đóng/Mở** => kiểm tra **ID** của client nếu trùng với thông tin gửi tới thì thực hiện **Đóng/Mở** form client.

Class **MovieStatistics**:

- **Nội dung**: Đối tượng dùng để **lưu thông tin** được người dùng chọn => nhằm thống kê doanh thu, xếp hạng phim khi xuất ra **file output**.

Class **ThongTin**:

- **Nội dung**: Đối tượng nhận các thông tin khởi tạo mặc định cho rạp.

Class **Hall:**

**- Nội dung:** Đối tượng khởi tạo **ghế và trạng thái** ghế trong 1 rạp (mặc định là **false** - ghế chưa được chọn).

List<**ThongTin**> **my\_list**:  
- **Nội dung**: Tạo list cho mỗi phim được nhập vào trước khi lưu **file input**.

Dictionary<**string, Hall**> **my\_hall**:

- **Nội dung**: Nhằm quản lý nhiều rạp và trạng thái ghế tương ứng vào rạp đó.

Hàm **Lưu**:

- **Nội dung**: **Serialize** thông tin (mảng **ThongTin**) ở screen sau khi add và lưu vào file input.

Button **Lưu**:

- **Nội dung**: gọi hàm **Lưu**.

Hàm **Deserialize**:  
- **Nội dung**: **Deserialize** thông tin trước khi lấy thông tin từ file và cập nhật lên giao diện

Hàm **IsCommaSeparated**:

- **Nội dung**: Kiểm tra thông tin số **rạp chiếu** được nhập từ người dùng với **rằng buộc** là nếu số lượng rạp nhiều hơn 1 thì phải ngăn cách bằng **dấu phẩy**.

Button **Add**:

- **Nội dung**: Kiểm tra thông tin được nhập và add vào screen để chuẩn bị cho việc lưu trữ.

- **Trình bày**:

* Kiểm tra dấu phẩy bằng **IsCommaSeparated(rapChieu\_box.Text)**
* Áp dụng **Regular Expression** cho **Giá vé** **System.Text.RegularExpressions.Regex.IsMatch(giaVe\_box.Text, @"^[1-9]\d\*$") – Giá vé** phải là số không bắt đầu từ 0 và không chứa ký tự.
* Kiểm tra đối chiếu với các thông tin đã có trong **Screen** để không xảy ra hiện tượng trùng lặp => nếu không có **trùng lặp** xảy ra thì dữ liệu được add vào **my\_list** trước khi lưu.

Button **layInfo**:

- **Nội dung**: Deserialize dữ liệu từ file input và cập nhật thông tin lên giao diện

ComboBox **phimSelection**:

- **Nội dung**: Clear các giá trị ghế được tick trước đó khi chọn phim khác và cập nhật giá vé cơ sở đối với đối với mỗi phim được chọn.

ComboBox **chonRap**:

- **Nội dung**: Làm sạch các giá trị ghế được **tick** trước đó khi chuyển sang **rạp mới** và gửi thông tin đến **server** để kiểm tra rạp hiện tại, nếu không có trong **cơ sở dữ liệu** => thực hiện câu lệnh **SQL** để tạo rạp mới trong cơ sở dữ liệu.

- **Trình bày**: gọi hàm **Send\_ThongTin()** đã được định nghĩa (hàm được set bất đồng bộ để không ảnh hưởng đến việc cập nhật giao diện).

Event **SeatSelect**:

- **Nội dung**: Kiểm tra xem các ghế chọn đã được đặt hay chưa, nếu đã được đặt rồi thì khôi phục lại trạng thái **unchecked**

**- Trình bày:** gọi hàm **IsSeatOccupiedInDatabase()** để check giá trị trong database.

Hàm **IsSeatOccupiedInDatabase**

- **Nội dung**: chọn các seat trong **database** dựa trên tên rạp và ghế được truyền vào, sau đó kiểm tra => nếu ghế đã được đặt (**IsOccupied** = 1) thì báo lỗi.

Button **Đặt Vé**:

- **Nội dung**: Hiển thị thông tin mua của người dùng (tên, phim đã chọn, số ghế đã chọn, tổng số tiền cần thanh toán), gửi thông tin đặt vé đến server.

- **Trình bày**:

* Sử dụng lệnh **If** để kiểm tra người dùng đã chọn đủ thông tin hay không.
* Tính toán giá vé dựa trên chỗ ngồi **đã chọn** => Sử dụng vòng **for**, sau đó tính toán số vé **còn lại**.
* **Send\_datVe()**: Gửi thông tin đặt vé đến server để cập nhật giá trị ghế.

Button **Xuất Thông Tin**:

- **Nội dung**: Hiển thị thông tin của phim đã mua (tên phim, số lượng vé bán ra, số lượng vé tồn, tỉ lệ vé bán ra, doanh thu, xếp hạng phòng vé).

- **Trình bày**:

* Sắp xếp thứ tự thông tin phim lưu vào file dựa trên doanh thu được tính toán từ mỗi lần mua vé => chuyển thành **List** và lưu vào biến **sortedStatistics.**
* Ghi vào file **output** các thông tin theo đề yêu cầu với mỗi đối tượng **MovieStatics**.
* Thanh **Progress Bar** giúp người dùng theo dõi tình trạng xuất file.

void **Bai4\_FormClosed()**:

- **Nội dung**: Đóng kết nối và kiểm tra có thread nào chạy nền hay không.

void **StartServer()**:

- **Nội dung**: Chạy server, nhận địa chỉ từ bất kỳ IP nào trong nội mạng.

Hàm **Receive()**:

- **Nội dung**: Lắng nghe các data được gửi đến từ phía client, sau đó thực hiện **Deserialize** dưới dạng **MovieTicket** tưởng ứng với định dạng được gửi đi sau đó bắt đầu xử lý dữ liệu.

- **Trình bày**:

* **text.StartsWith("1")**: Kiểm tra dữ liệu truyền vào có phải là dữ liệu khởi tạo rạp được gửi từ phía client hay không (hàm **Send\_ThongTin**). Nếu có thì kiểm tra xem rạp đã tồn tại hay chưa **IsHallInDatabase(ticketInfo)** rồi thực hiện thêm rạp **InsertSeatAvailability(ticketInfo).**
* Nếu **text.StartsWith("2")** thì dữ liệu gửi đến sẽ được coi như là dữ liệu nhằm update trạng thái rạp từ **0 => 1** (Occupied). Sau đó thêm thông tin của vé vừa được mua vào **screen.**
* Nếu **text.StartsWith("DONGMO")** thì dữ liệu được gửi đến sẽ được hiểu là **server** có nhiệm vụ **broadcast** đến các **client khác** về lệnh **Đóng/Mở,** nếu **ID** của **client** nào **trùng** với thông tin trong lệnh thì client đó sẽ bị **đóng hoặc mở**.

Hàm **IsHallInDatabase()**:  
- **Nội dung**: Khởi tạo kết nối với **database**. Kiểm tra dữ liệu đã có trong cơ sở dữ liệu hay chưa.

Hàm **InsertSeatAvailability()**:

- **Nội dung**: Thực hiện không đồng bộ việc thêm rạp vào trong cơ sở dữ liệu với query đã được định nghĩa.

Hàm **UpdateSeatAvailability():**

**- Nội dung:** Khởi tạo kết nối với **database**. Cập nhật trạng thái chỗ được đặt trong **database** với những ghế có trong data **foreach (string seat in data.Seats)**.

Hàm **Deserialize()**:  
- **Nội dung**: **Deserialize** thông tin được nhận (**JSON Format**).

void **server\_FormClosed()**:

- **Nội dung**: Thực hiện việc ngắt kết nối khi form được đóng.

Form Đăng Nhập của **SuperUser**:

Void **NotAdmin\_LinkClicked()**:

**- Nội dung**: Đóng form nếu không phải Admin (dùng function **Close()**).

Void **Login\_Click()**:  
- **Nội dung**: Gửi lệnh Đóng/Mở form tới server.

- **Trình bày**:

* Kiểm tra thông tin **Username** và **Password** từ input được nhập bởi người dùng.
* Nếu thông tin chính xác thì form **SuperUser** được mở ra, nếu không thì in ra màn hình thông báo lỗi.

Form **DieuHuong(**):

- **Nội dung**: Mở các form để thao tác với chương trình

- **Trình bày**:

* Mở **server** => mở các **client** quầy vé.
* Nếu là **SuperUser** thì click vào LinkLabel để chuyến hướng đến form đăng nhập của **SuperUser**.

Form **SuperUser()**:

Void **ConnectToServer()**:

- **Nội dung**: Khai báo địa chỉ IP đích, client, kết nối đến Server.

- **Trình bày**:

- Khai báo địa chỉ đích của server **127.0.0.1 – địa chỉ LoopBack**, có thể đổi sang địa chỉ khác ví dụ **192.168.1.15** nếu server ở máy tính khác.

Public class **ThongTinDongMo**:

- **Nội dung**: Gửi đi ID client cần thực hiện lệnh Đóng/Mở

- **Trình bày**:

* **ID**: ID của client cần thực hiện lệnh Đóng/Mở.
* **Header**: Lệnh đóng hoặc mở dựa trên button người dùng chọn.

Void **SendToSV()**:

- **Nội dung**: Hàm gửi đi thông tin tới **Server**.

- **Trình bày**:

* Gán thông tin input từ người dùng vào đối tượng **ThongTinDongMo**.
* Thêm Header **DONGMO** để phân biệt với các thông tin gửi đi khác.
* Chuyển sang dạng **JSON** cho thông tin sau đó mã hóa **UTF8** rồi gửi đi.

**YÊU CẦU CHUNG**

1. Đánh giá

* Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
* Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
* Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả *(nếu có)*; giải thích cho quan sát *(nếu có)*.
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

1. Báo cáo

* Nộp file .docx. Tập trung vào nội dung, giải thích.
* Nội dung trình bày bằng Font chữ Cambria hoặc Times New Roman (*tuy nhiên, phải chuyển đổi hết báo cáo này sang 1 font chữ thống nhất*) – cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: Mã lớp-LabX\_MSSV1\_MSSV2. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: NT106.O21.1-Lab01\_25520001\_25520002

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài thực hành.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

**HẾT**