**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**BÁO CÁO ĐỒ ÁN 1**

**PHẦN MỀM CHẤM FILE CPP VÀ CS**

**GVHD : ThS.Nguyễn Trần Thi Văn**

**SVTH : Nguyễn Duy Lâm 16110134**

**Đặng Nguyễn Hồng Kha 16110112**

**Tp. Hồ Chí Minh, tháng 11 năm 2018**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(ký và ghi họ tên)

…………………….....

MỤC LỤC

[MỤC LỤC 3](#_Toc438545396)

[Danh mục các bảng 4](#_Toc438545396)

[Danh mục các hình 5](#_Toc438545396)

**I**. [**MÔ TẢ PROJECT** 6](#_Toc438545396)

1. Lí do xây dựng phần mềm. [6](#_Toc438545396)

2.Mục đích sử dụng phần mềm [6](#_Toc438545396)

[3.Use case diagram](#_Toc438545396) 6

4.Mô tả use case diagram [7](#_Toc438545396)

[**II. THỰC HIỆN** 1](#_Toc438545396)0

[1.Thiết kế giao diện 1](#_Toc438545396)0

[2.Thiết kế lớp](#_Toc438545396) 12

[3.Thiết kế các phương thức](#_Toc438545396) 13

[**III.ĐÁNH GIÁ VÀ PHÂN CÔNG**](#_Toc438545396) 26

[1.Phân công](#_Toc438545396) 26

[2.Đánh giá 26](#_Toc438545396)

**DANH MỤC CÁC BẢNG**

Bảng 1: Mô các các Actor

Bảng 2: Mô tả các Use Case

Bảng 3: Mô tả thiết kế giao diện

Bảng 4: Mô tả các lớp

Bảng 5: Mô tả các phương thức

Bảng 6: Phân công các thành viên

**DANH MỤC CÁC HÌNH**

Hình 1:Use Case Diagrama

Hình 2: Form chức năng

Hình 3:Form chấm bài

Hình 4: Form hiển thị kết quả

Hình 5: Form giao diện tổng quan

**I. Mô tả project**

1. **Lý do xây dựng phần mềm**

* Từ thực tiễn cho thấy việc khi giáo viên chấm code của sinh viên thủ công bằng cách build và nhập các test case sẽ gây ra việc mất nhiều thời gian và công sức. Chính vì vậy, phần mềm chấm bài C++ và C# được xây dựng để giúp việc chấm bài trở nên nhanh chóng và dễ dàng hơn chỉ gom gọn trong vài cái click chuột.

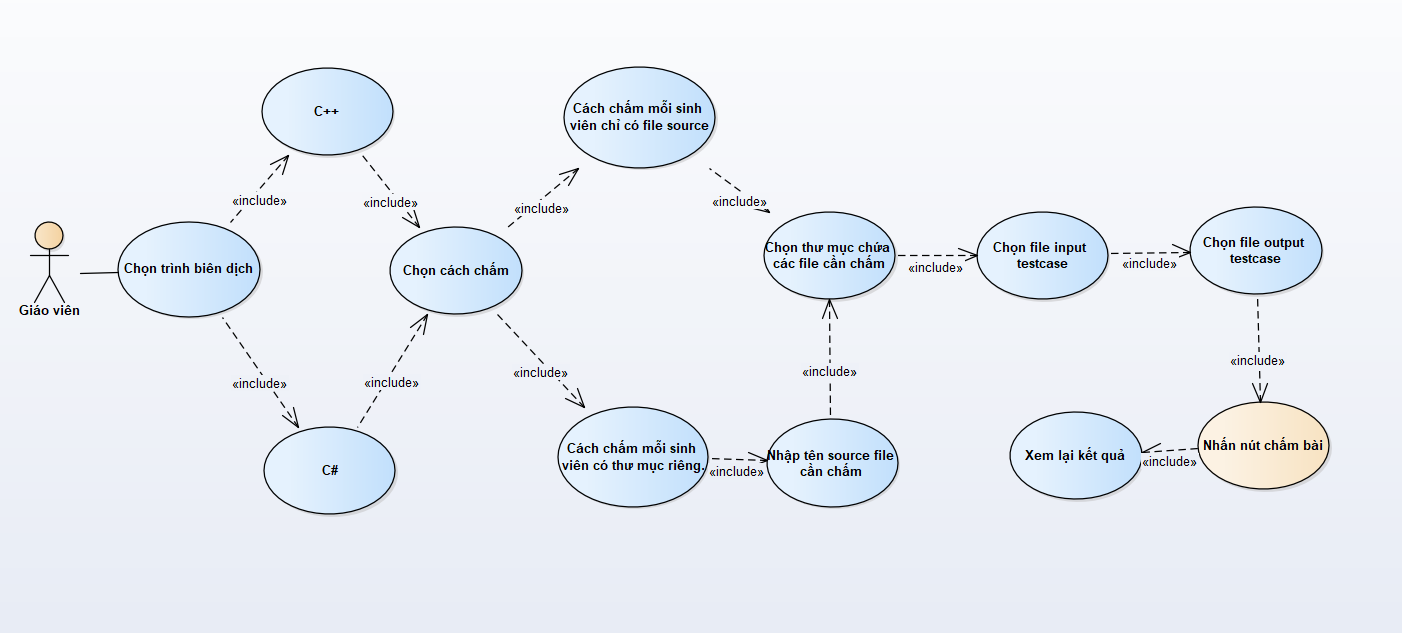
1. **Mục đích của phần mềm**

* Phần mềm dùng để build các file .cpp của C++ và file .cs của C# thành exe và chấm điểm.
* Phần mềm hổ trợ chấm điểm ở hai ngữ cảnh thực tế sau:   
  + Mỗi thí sinh nộp một cho giáo viên chỉ 1 file source. Thông tin của bài thi như số báo danh và tên thí sinh là tên file source.

+ Mỗi thí sinh nộp bài cho giáo viên nhiều file source, những file source được chứa trong một folder và thông tin của thí sinh là tên folder.

* Phần mềm sẽ chấm bài và hiển thị thông tin của tất cả thí sinh và điểm lên của họ màn hình hiển thị.
* Giáo viên có thể xem lại tất cả kết quả mà phần mềm build được của bất kỳ thí sinh nào.

1. **Use case diagram**



Hình 1: Use case diagram

1. **Mô tả use case diagram**

|  |  |  |
| --- | --- | --- |
| TT | Tên Actor | Ý nghĩa |
|  | Giáo viên | Được sử dụng tất cả các chức năng chấm điểm |

Bảng 1: Mô tả các Actor

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
|  | Chọn trình biên dịch | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Chọn một trong 2 trình biên dịch C++ hoặc C#  **Các bước thực hiện**: Click vào chon mục C++ hoặc C# |
|  | C++ | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Chọn sử dụng trình biên dịch C++ để chấm bài.  **Các bước thực hiện**: : Click vào chon mục C++ |
|  | C# | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Chọn sử dụng trình biên dịch C# để chấm bài.  **Các bước thực hiện**: : Click vào chon mục C# |
|  | Chọn cách chấm | **Actor:** Giáo viên.  **Mô tả tóm tắt:**chọn cách chấm thích hợp với cách nộp bài  **Các bước thực hiện**: : Click vào chon mục Chỉ source file hoặc Mỗi |
|  | Mỗi sinh viên chỉ có File Source | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Cách chấm này sử dụng khi sinh viên nộp bài là một source file, thông tin sinh viên là tên của source file đó (VD: HongKha 16110112.cpp)  **Các bước thực hiện**: click vào Mỗi sinh viên chỉ có File Source |
|  | Mỗi sinh viên có thư mục riêng | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Cách chấm này sử dụng khi sinh viên nộp một hoặc nhiều bài tên có cấu trúc. (BAI1.cpp, BAI2.cpp, …) và tất cả các bài của một sinh viên được lưu trong 1 thư mục. Tên thư mục là thông tin của sinh viên.  **Các bước thực hiện**: Click vào Mỗi sinh viên có thư mục riêng |
|  | Nhập tên Source file cần chấm | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Do khi chấm theo cahcs mỗi Source file có thư mục riêng  **Các bước thực hiện**:Click vào Thư Mục Chứa Source.  **Yêu cầu trước khi thực hiện:**Phải chọn đúng thư mục. |
|  | Chọn địa chỉ thư mục cần chấm | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Chọn địa chỉ thư mục cần chấm.  **Các bước thực hiện**: Click vào button Thư mục chứa code.  **Yêu cầu trước khi thực hiện:**Phải là thư mục mẹ chứa các source hoặc chứa các thư mục chứa source. |
|  | Đường dẫn input | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Dùng để dẫn đến input testcase (nếu có)  **Các bước thực hiện** : Click vào button Đường dẫn file input. |
|  | Đường dẫn file output | **Actor:** Giáo viên.  **Mô tả tóm tắt:**Dùng để dẫn đến output của testcase (nếu có)  **Các bước thực hiện :** Click vào button Thư mục chứa code. |
|  | Chấm Bài | **Actor:** Giáo viên  **Mô tả tóm tắt:**Phần mềm sẽ bắt đầu build và chấm bài và xuất tất cả ra kết quả ra màn hình.  **Các bước thực hiện**: Click vào button Chấm bài |
|  | Xem lại kết quả | **Actor:** Giáo viên  **Mô tả tóm tắt:**Xem lại kết quả của từng cá nhân bằng cách nhập số thứ tự.  **Các bước thực hiện**: Nhập thứ tự. Click vào button Xem |

Bảng 2 : Mô tả Use case diagram

# Thực hiện

1. **Thiết kế giao diện**

|  |  |  |
| --- | --- | --- |
| TT | Màn hình | Mục đích chính của màn hình |
| 1. | Form Chọn các chức năng | Chọn các chức năng mình cần xử lý. |
| 2. | Form Dùng Để Chấm Bài | Khi chọn các chức năng ở trên tương ứng thì bắt đầu Chấm Bài |
| 3. | Form hiển thị kết quả và chấm bài | Hiện thị kết quả các source file được chấm. |
| 4. | Giao diện tổng quan: |  |

Bảng 3 : Mô tả thiết kế giao diện

1. **Thiết kế các lớp**

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Tên các SV phụ trách viết  (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ ngồn nào). | Mục đích chính của lớp trong chương trình |
|  | ExecuteShellComand | Đặng Nguyễn Hồng Kha | Lớp đảm nhiệm các thao tác với command prompt. |
|  | LISTDIR | Đặng Nguyễn Hồng Kha | Lớp đảm nhiệm đọc cấu trúc của cây thư mục. Phục vụ cho tính năng chấm nhiều source file đặt chung một folder. |
|  | ReadFile | Nguyễn Duy Lâm | Lớp đảm nhiệm các thao tác đọc và ghi file văn bản. Hổ trợ cho tính năng đọc và so sánh kết quả của file chấm và các bộ test case |
|  | SyncPipe | http://chillyfacts.com/run-command-prompt-cmd-commands-from-java/ | Lớp hổ trợ cần bắt buộc cho lớp ExcuteShellCommand |
|  | TakeFilePath | Nguyễn Duy Lâm | Lớp hổ trợ các thao tác lấy địa chỉ tuyệt đối của một file khi dùng JFileChooser |
|  | FrmMain | Đặng Nguyễn Hồng Kha | Bố trí giao diện và bắt các sự kiện hổ trợ tính năng chấm bài |

Bảng 4 : Thiết kế các lớp

1. **Thiết kế các phương thức**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức | Tên các SV phụ trách viết | Mục đích | Tên file, số thứ tự dòng chứa khai báo |
| 1. | public void RunComand(String[] strCommand, int numberOfCommand)  **Input:** strCommand, numberOfCommand  **Output:** không có  **Pseudo code:**  -Gọi cmd.  -***Lặp*** từ 0 < **numberOfCommand**  -Ghi các lệnh **strCommand[i]** vào màn hình command prompt.  -Thực thi các lệnh | Đặng Nguyễn Hồng Kha | Gọi command prompt, nhập vào command những dòng lệnh gọi trình biên dịch, build file cpp/cs thành exe và xuất ra file cpp. | ExecuteShellComand.java  (21) |
| 2. | public String[] takeDirectoryTree(String path, String Tail)  **Input:** path, Tail  **Output:** PATH  **Pseudo code:**  -Lấy **listFiles** ở địa chỉ **path**  -***Nếu*** không có **tail *thì***lấy tên toàn bộ thư mục có trong path.  -***Nếu*** có **tail** ***thì*** lấy tên toàn bộ file có đuôi là **tail.**  - ***Trả về*** mảng chứa các tên cần lấy. | Nguyễn Duy Lâm | Trả về danh sách tên tất cả các file trong cây thư mục có đuôi là tail (được truyền vào). Phục vụ cho công việc lấy tên của tất cả các file cpp/cs có trong thư mục | LISTDIR.java  (68) |
| 3. | public String readTextFile(  String strPATH)  **Input:** strPATH  **Output:** allLine  **Pseudo code:**  -***Mở file*** ở địa chỉ **strPATH**  -***Cộng dồn*** nội dung từng dòng trong file cho đến cuối file.  - ***Trả về*** chuỗi nội dung của file. | Nguyễn Duy Lâm | Trả về toàn bộ nội dung của một file văn bản. | ReadFile.java  (20) |
| 4 | public String [] readTextFileToArray(String strPATH)  **Input:** strPATH  **Output:** kqPATH  **Pseudo code:**  -***Mở*** *file* ở địa chỉ **strPATH**  -***Lần lượt*** lưu nội dung từng dòng trong file vào mảng **kqPATH** cho đến cuối file.  - ***Trả về*** mảng **kqPATH**. | Nguyễn Duy Lâm | Trả về một mảng String toàn bộ nội dung của một file văn bản. Mỗi phần tử mảng là một dòng của văn bản. | ReadFile.java  (53) |
| 5 | public String takeFolderDerectory()  **Input:** không có.  **Output:** chooser.getSelectedFile().toString()  **Pseudo code:**  -***Mở***hộp thoại thư mục bằng **JfileChooser**.  -***Nếu*** chọn “ **Choose**” ***thì*** ***trả về*** địa chỉ thư mục đã chọn. | Nguyễn Duy Lâm | Lấy địa chỉ tuyệt đối của một Folder khi sử dụng JfileChooser. | TakeFilePath.java  (25) |
| 6 | public String takeFileDerectory (String Filter, String Depcription, String CurrentDerectory)  **Input:** Filter, Depcription, CurrentDerectory  **Output:** chooser.getSelectedFile().toString();  **Pseudo code:**  -***Mở***hộp thoại thư mục bằng **JfileChooser** ở vị trí **CurrentDerectory**.  -***Lọc ra*** các file có đuôi là **Filter** và đặt mô tả đuôi là **Depcription**.  -***Nếu*** chọn “ **Choose**” ***thì*** *trả về* địa chỉ file đã chọn. | Nguyễn Duy Lâm | Lấy địa chỉ tuyệt đối của một file khi có chọn filter.  CurrentDerectory là biến dùng để hổ trợ việc khi mở lên lần thứ 2. JfileChooser sẽ mở lên ở thư mục đã chọn trước đó. | TakeFilePath.java  (42) |
| 7 | public String takeFileDerectory()  **Input:** không có  **Output:** chooser.getSelectedFile().toString();  **Pseudo code:**  -***Mở***hộp thoại thư mục bằng **JfileChooser.**  -***Nếu*** chọn “ **Choose**” ***thì*** *trả về* địa chỉ file đã chọn. | Nguyễn Duy Lâm | Lấy địa chỉ tuyệt đối của một file không có filter. | TakeFilePath.java  (67) |
| 8 | pritive void btnChamBaiClick()  **Input:** không có  **Output:** không có  **Pseudo code:**  -***Nếu*** có input của testcase ***thì*** ***đọc file*** input vào chương trình.  -***Nếu*** có output của testcase ***thì đọc file*** output vào chương trình.  -**Nếu** cách chấm bài là chỉ có source file thì:  + **Đọc cây thư mục** để lấy tên tất cả các source file.  + **Ứng với** từng tên file có được. **Gọi hàm chấm bài** để build file.  + **Đọc file** kết quả mà file build ra lưu vào **ExecuteReport**.  + **Hiển thị** ra kết quả từ **ExecuteReport** ra màn hình.  -Nếu cách chấm bài là Mỗi source có thư mục riêng:  + **Đọc cây thư mục** để lấy tên tất cả các thư mục có chứa source file.  **+Ứng với** từng thư mục. **Gọi hàm chấm bài** để build file.  + **Hiển thị** ra kết quả từ mảng toàn cục chứa kết cả chấm bài **ExecuteReport** ra màn hình. | Đặng Nguyễn Hồng Kha | Sử dụng khi người dùng chọn compiler C++, hàm chạy khi người dùng nhấn nút chấm bài.  Hàm sẽ chuẩn bị load các testcase (nếu có)  Phân hoạch cách chấm bài và gọi hàm runChamBai để chấm bài lần lượt cho từng file cpp. | frmMain.java  (661) |
| 9 | private void runChamBai()  **Input:** không có  **Output:** không có  **Pseudo code:**  -***Khởi tạo*** mảng **ExecuteResult** chứa kết quả của bài chuẩn bị chấm.  -***Nếu*** là lần build bài đầu tiên ***thì*** gọi **excuteCommand** để đưa các lệnh build cpp thành exe xuống command prompt để build và nhập testcase.  -***Nếu*** không là lần build đầu tiên ***thì*** tắt cờ để ***excuteCommand*** không phải build lại thành file exe mà chỉ nhập textcase.  -***Đọc file*** kết quả sau khi build và lưu vào mảng toàn cục **ExecuteResult**.  -***So sánh*** kết cả sau khi build được bằng input testcase với bộ output testcase để tính điểm. | Đặng Nguyễn Hồng Kha | Hàm có nhiệm vụ đưa lần lượt các bộ testcase tương ứng để khi command được khởi chạy.  Lấy kết quả của mỗi lần execute và so sánh với bộ testcase để chấm điểm. | frmMain.java  (611) |
| 10 | private void excuteCommand(String otherCommand, Boolean isFistTime)  **Input:** otherCommand, isFistTime  **Output:** không có  **Pseudo code:**  -***Khởi tạo*** đối tượng **ExecuteShellComand**  -Khởi tạo mảng một chiều **strDirectToFoloder:**  + **strDirectToFoloder[0]** chứa: Đường dẫn ổ đĩa của file cần build.  + **strDirectToFoloder[1]** chứa: lệnh cd đến đường dẫn file cần build.  + **strDirectToFoloder[2]** chứa: lệnh build file thành exe nếu **isFirstTime** = true hoặc rỗng nếu **isFirstTime** = false;  + **trDirectToFoloder[3]** chứa: lệnh chạy file exe vừa build được và lưu kết quả ra 1 file txt.  + **trDirectToFoloder[4]** chứa: **otherCommand** tức bộ input testcase nếu có.  -Gọi phương thức **runCommand** của lớp **ExecuteShellComand** để thực thi việc chấm bài. | Đặng Nguyễn Hồng Kha | Hàm có nhiệm vụ nhận các testcase từ runChamBai và đưa xuống command prompt khi gọi các lệnh thực thi build file và chấm bài. (Các lệnh build file do trình biên dịch TDM-GCC-64 hổ trợ) | frmMain.java  (563) |
| 11 | private void btnChamBaiClick\_1()  **Input:** không có  **Output:** không có  **Pseudo code:**  -***Nếu*** có input của testcase ***thì*** ***đọc file*** input vào chương trình.  -***Nếu*** có output của testcase ***thì đọc file*** output vào chương trình.  -**Nếu** cách chấm bài là chỉ có source file thì:  + **Đọc cây thư mục** để lấy tên tất cả các source file.  + **Ứng với** từng tên file có được. **Gọi hàm chấm bài** để build file.  + **Đọc file** kết quả mà file build ra lưu vào **ExecuteReport**.  + **Hiển thị** ra kết quả từ **ExecuteReport** ra màn hình.  -Nếu cách chấm bài là Mỗi source có thư mục riêng:  + **Đọc cây thư mục** để lấy tên tất cả các thư mục có chứa source file.  **+Ứng với** từng thư mục. **Gọi hàm chấm bài** để build file.  + **Hiển thị** ra kết quả từ mảng toàn cục chứa kết cả chấm bài **ExecuteReport** ra màn hình. | Đặng Nguyễn Hồng Kha | Sử dụng khi người dùng chọn compiler C#, hàm chạy khi người dùng nhấn nút chấm bài.  Hàm sẽ chuẩn bị load các testcase (nếu có)  Phân hoạch cách chấm bài và gọi hàm runChamBai\_1 để chấm bài lần lượt cho từng file cs. | frmMain.java  (803) |
| 12 | private void runChamBai\_1()  **Input:** không có  **Output:** không có  **Pseudo code:**  -***Khởi tạo*** mảng **ExecuteResult** chứa kết quả của bài chuẩn bị chấm.  -***Nếu*** là lần build bài đầu tiên ***thì*** gọi **executeCommandWithDotNet** để đưa các lệnh build cpp thành exe xuống command prompt để build và nhập testcase.  -***Nếu*** không là lần build đầu tiên ***thì*** tắt cờ để ***executeCommandWithDotNet*** không phải build lại thành file exe mà chỉ nhập textcase.  -***Đọc file*** kết quả sau khi build và lưu vào mảng toàn cục **ExecuteResult**.  -***So sánh*** kết cả sau khi build được bằng input testcase với bộ output testcase để tính điểm. | Đặng Nguyễn Hồng Kha | Hàm có nhiệm vụ đưa lần lượt các bộ testcase tương ứng để khi command được khởi chạy.  Lấy kết quả của mỗi lần execute và so sánh với bộ testcase để chấm điểm. | frmMain.java  (945) |
| 13 | private void executeCommandWithDotNet (String otherCommand, Boolean isFistTime)  **Input:** otherCommand, isFistTime  **Output:** không có  **Pseudo code:**  -***Khởi tạo*** đối tượng **ExecuteShellComand**  -Khởi tạo mảng một chiều **strDirectToFoloder:**  + **strDirectToFoloder[0]** chứa: Đường dẫn ổ đĩa của file cần build.  + **strDirectToFoloder[1]** chứa: lệnh cd thư mục .NETFrameWork trong hệ thống.  + **strDirectToFoloder[2]** chứa: lệnh build file thành exe nếu **isFirstTime** = true hoặc rỗng nếu **isFirstTime** = false;  + **trDirectToFoloder[3]** chứa: lệnh chạy file exe vừa build được và lưu kết quả ra 1 file txt.  + **trDirectToFoloder[4]** chứa: **otherCommand** tức bộ input testcase nếu có.  -Gọi phương thức **runCommand** của lớp **ExecuteShellComand** để thực thi việc chấm bài. | Đặng Nguyễn Hồng Kha | Hàm có nhiệm vụ nhận các testcase từ runChamBai\_1 và đưa xuống command prompt khi gọi lệnh thực thi build file và chấm bài.(Các lệnh build file do trình biên dịch .NET hổ trợ) | frmMain.java  (586) |

Bảng 5 : Mô tả các giao thức

1. **Phân công và đánh giá**
2. **Phân công**

|  |  |  |
| --- | --- | --- |
| Tên SV | Đóng góp | Công việc |
| Đặng Nguyễn Hồng Kha |  |  |
| Nguyễn Duy Lâm |  |  |

Bảng 5: Phân công các thành viên

1. **Đánh giá**:

*Ưu điểm:*

Nhóm 1 đã hoàn thành 100% việc xây dựng code project. Bao gồm chấm file cpp và cs đáp ứng các tính năng:

+ Chấm bài gồm tập hợp những các sinh viên làm nhiều bài, mỗi bài đặt tên có quy tắc riêng (VD: Bai1.cpp, Bai\_2.cpp,…) và được gom vào 1 thư mục chính.

+ Chấm 1 hoặc nhiều source file đặt chung 1 thư mục. (Không cần đặt tên theo quy tắc). Chức năng này áp dụng cho trường hợp sinh viên nộp 1 file cpp duy nhất, tên file không cần theo quy tắc nào, người dung chỉ cần gom tất cả các file cpp vào chung 1 thư mục để chấm chung. Phần mềm giải quyết vấn đề bằng cách đọc cây thư mục.

*Khuyết điểm:*

Vấn đề liên quan đến compiler của GCC và quyền admin khi truy cập vào thư mục .NET Framework không thể giải quyết được vì bản chất vấn đề nằm ở chính trình biên dịch. Nên cách giải quyết của phần mềm là đưa ra các hướng dẫn, mô tả cụ thể cho người dùng các tính năng.