**Project Timeline**

## **Ngày 1-2: Phân Tích Yêu Cầu và Thiết Kế Các Test Case**

1. **Phân tích yêu cầu:**
   * **User:** Xác định vai trò của User (Admin, Contributor, Reader) và cách mỗi vai trò ảnh hưởng đến quyền truy cập.
   * **Drive:** Mỗi User có thể sở hữu nhiều Drive; mỗi Drive có thể chứa nhiều Folder gốc.
   * **Folder:** Một Folder có thể chứa nhiều Sub Folder và File, quyền có thể được gán cho các Folder.
   * **File:** File thuộc về một Folder cụ thể và có thể có quyền tương tự như Folder.
2. **Thiết kế test case:**
   * **Test Case cho Admin:**
     + Chia sẻ quyền với User khác cho các Folder và File.
     + Quản lý quyền cho các Sub Folder và File.
   * **Test Case cho Contributor:**
     + Thực hiện thao tác thêm, sửa, xóa Folder và File.
   * **Test Case cho Reader:**
     + Chỉ xem File và Folder mà không thể thực hiện bất kỳ thay đổi nào.

**Kết quả mong đợi:** Các test case cho từng vai trò và chức năng đã được thiết kế rõ ràng.

## **Ngày 3: Xây Dựng Các Lớp Đối Tượng Cơ Bản và Viết Unit Test**

1. **Viết Unit Test:**
   * **Test Case cho lớp User:**
     + Tạo User với nhiều Drive.
     + Gán và kiểm tra quyền cho User trên các Folder và File.
   * **Test Case cho lớp Drive:**
     + Thêm và xóa Drive, kiểm tra việc chứa nhiều Folder gốc.
   * **Test Case cho lớp Folder:**
     + Thêm, sửa, xóa Folder; kiểm tra quyền và cấu trúc Sub Folder.
   * **Test Case cho lớp File:**
     + Thêm, sửa, xóa File trong Folder; kiểm tra quyền truy cập File.
2. **Triển khai các lớp cơ bản:**
   * **User Class:** Đại diện cho User, có danh sách Drive và quyền.
     + **Thuộc tính:** List<Drive> Drives, List<Permission> Permissions.
     + **Phương thức:** AddDrive(), GrantPermission(), RemovePermission(), HasPermission().
   * **Drive Class:** Đại diện cho Drive, chứa các Folder gốc.
     + **Thuộc tính:** List<Folder> RootFolders.
     + **Phương thức:** AddFolder(), RemoveFolder(), AddFile(), RemoveFile().
   * **Folder Class:** Đại diện cho Folder, chứa Sub Folder và File.
     + **Thuộc tính:** List<Folder> SubFolders, List<File> Files, List<User> UsersWithPermissions.
     + **Phương thức:** AddFolder(), RemoveFolder(), AddFile(), RemoveFile().
   * **File Class:** Đại diện cho File, thuộc về một Folder cụ thể.
     + **Phương thức:** ModifyFile().

**Kết quả mong đợi:** Các lớp cơ bản được triển khai và các Unit Test cho chúng đã pass.

## **Ngày 4: Xây Dựng Hệ Thống Quản Lý Quyền Truy Cập**

1. **Viết Unit Test cho quản lý quyền:**
   * **Test Case cho GrantPermission:** 
     + Gán quyền cho User trên Folder và File.
   * **Test Case cho RemovePermission:** 
     + xóa quyền cho User trên Folder và File.
   * **Test Case cho HasPermission:**
     + Kiểm tra quyền của User với các hành động cụ thể.
   * **Test Case cho HasOwnerPermission:** 
     + Gán quyền cho User trên Folder và File.
2. **Triển khai các phương thức quản lý quyền:**
   * **GrantPermission:** Gán quyền cho User trên Folder và File.
     + **Phương thức:** GrantPermission(User user, Folder folder, PermissionRole role).
   * **HasPermission:** Kiểm tra quyền của User.
     + **Phương thức:** HasPermission(User user, Folder folder, PermissionRole role).

**Kết quả mong đợi:** Hệ thống quản lý quyền hoạt động theo yêu cầu và các Unit Test liên quan pass.

## **Ngày 5: Triển Khai Các Vai Trò Admin, Contributor, Reader**

1. **Viết Unit Test cho các vai trò:**
   * **Admin:**
     + Chia sẻ quyền và quản lý quyền truy cập.
     + **Test Case:** Admin chia sẻ quyền cho User khác, kiểm tra quyền truy cập của User.
   * **Contributor:**
     + Thực hiện các thao tác thêm, sửa, xóa.
     + **Test Case:** Contributor thêm, sửa, xóa Folder và File.
   * **Reader:**
     + Chỉ xem, không thể sửa đổi.
     + **Test Case:** Reader xem File và Folder, không thể thực hiện thay đổi.
2. **Triển khai các lớp và phương thức để hỗ trợ các vai trò:**
   * **Admin Role:**
     + **Phương thức:** SharePermissions(User user, Folder folder, PermissionRole role).
   * **Contributor Role:**
     + **Phương thức:** ModifyContent(Folder folder, File file).
   * **Reader Role:**
     + **Phương thức:** ViewContent(Folder folder, File file).

**Kết quả mong đợi:** Các vai trò và quyền được phân định rõ ràng và các Unit Test cho vai trò đã pass.

## **Ngày 6: Kiểm Thử Hệ Thống và Hoàn Thiện**

1. **Chạy toàn bộ Unit Test:**
   * Đảm bảo tất cả các Unit Test đều pass.
   * Sửa lỗi nếu có test nào fail.
2. **Tối ưu hóa code:**
   * Xem xét lại và tối ưu hóa mã nguồn nếu cần thiết.

**Kết quả mong đợi:**

* + Hệ thống hoạt động chính xác theo yêu cầu và tất cả các Unit Test đều pass.

## **Ngày 7: Tinh Chỉnh và Chuẩn Bị Báo Cáo**

1. **Tinh chỉnh code:**
   * Hoàn thiện các phần còn lại và đảm bảo không còn lỗi.
2. **Chuẩn bị tài liệu:**
   * Tạo tài liệu hoặc báo cáo về hệ thống và quy trình kiểm thử.

**Kết quả mong đợi:** Hệ thống hoàn thiện và sẵn sàng nộp bài.

## **Đề bài**

Bạn được giao nhiệm vụ thiết kế và triển khai một hệ thống quản lý quyền truy cập cho cấu trúc Folder và File trên nhiều Drive bằng bất kỳ ngôn ngữ lập trình nào.

Mỗi User có thể sở hữu nhiều Drive, và mỗi Drive có thể chứa nhiều Folder và File. Mỗi User có thể được chỉ định các vai trò khác nhau, bao gồm:

**Admin**: Quản lý toàn bộ hệ thống và có thể chia sẻ quyền truy cập với User khác.

**Contributor**: Có thể thêm, chỉnh sửa và xóa Folder và File trong phạm vi được chỉ định.

**Reader**: Chỉ có quyền truy cập đọc và không thể thực hiện bất kỳ thay đổi nào.

1. **Yêu cầu:**

User có nhiều Drive

Một Drive có nhiều Folder và File.

Folder có nhiều Sub Folder và File

Sub Folder cũng có thể chứa nhiều Folder và File.

Xây dựng hệ thống quản lý quyền truy cập cho phép phân quyền dựa trên các vai trò như Quản trị viên, Người đóng góp, và Người đọc.

Triển khai các chức năng kiểm tra quyền truy cập, và cho phép thêm, chỉnh sửa và xóa Folder hoặc File dựa trên vai trò của User.

Quản trị viên có thể chia sẻ File, Folder, và Sub Folder:

* Nếu quản trị viên chia sẻ một Folder, thì User có quyền tương ứng cho Folder, Sub Folder, File...
* Nếu quản trị viên chia sẻ một File hoặc Sub Folder, User có hành vi tương tự nhưng không có quyền cho Folder cha.

1. **Phân tích và thiết kế hệ thống**

**Phân tích yêu cầu:**

Đầu tiên, xác định các đối tượng và vai trò trong hệ thống:

User (User): Mỗi User có thể sở hữu nhiều Drive và có thể có các vai trò khác nhau như Quản trị viên, Người đóng góp hoặc Người đọc cho các Folder hoặc File cụ thể.

Drive (Drive): Mỗi Drive chứa nhiều Folder gốc và có thể chứa các Sub Folder và File.

Folder (Folder): Một Folder có thể chứa các Sub Folder và File. Quyền truy cập có thể được gán cho User đối với các Folder này.

File (File): Một File thuộc về một Folder cụ thể và có thể có các quyền truy cập tương tự như Folder.

1. **Định nghĩa cấu trúc dữ liệu**

Bạn cần thiết kế các lớp để biểu diễn các đối tượng này:

Lớp User (User): Đại diện cho User, có danh sách các Drive và có thể có các quyền truy cập được gán.

Lớp Drive (Drive): Đại diện cho một Drive, chứa các Folder gốc.

Lớp Folder (Folder): Đại diện cho một Folder, có thể chứa các Sub Folder và File, và có danh sách User với các vai trò được gán khác nhau.

Lớp File (File): Đại diện cho một File, thuộc về một Folder cụ thể và cũng có thể có các quyền truy cập.

1. **Thiết kế vai trò và quyền truy cập**

Vai trò quyền truy cập (PermissionRole): Định nghĩa các vai trò như Quản trị viên, Người đóng góp và Người đọc.

Phương thức gán quyền truy cập (GrantPermission Method): Phương thức này dùng để gán vai trò cho User đối với một Folder hoặc File cụ thể.

Phương thức kiểm tra quyền truy cập (HasPermission Method): Phương thức này kiểm tra xem liệu một User có quyền thực hiện hành động dựa trên vai trò của họ hay không.

1. **Triển khai và kiểm thử**

Sau khi xây dựng các lớp và phương thức cần thiết, bạn cần viết các trường hợp kiểm thử để đảm bảo hệ thống hoạt động chính xác:

Vai trò Quản trị viên: Kiểm thử rằng User với quyền Quản trị viên có thể chia sẻ vai trò với User khác.

Vai trò Người đóng góp: Kiểm thử rằng User với quyền Người đóng góp có thể thêm, chỉnh sửa và xóa Folder và File.

Vai trò Người đọc: Kiểm thử rằng User với quyền Người đọc chỉ có thể xem và không thể chỉnh sửa nội dung.

1. **Kết luận:**

Bằng cách phân tích và thiết kế hệ thống như đã nêu, bạn sẽ xây dựng một hệ thống quản lý quyền truy cập mạnh mẽ và linh hoạt, đáp ứng nhu cầu đa dạng của User trong việc quản lý và truy cập Folder và File trên nhiều Drive. Hệ thống này sẽ đảm bảo rằng quyền truy cập được kiểm soát chặt chẽ và phù hợp với vai trò của User.

1. **Đừng quên khám phá thêm các trường hợp kiểm thử và viết unit test cho chúng.**

Các kịch bản ví dụ

Kịch bản 1: User Quản trị viên chia sẻ quyền truy cập

User: admin\_user có quyền Quản trị viên trên Root1.

Hành động: Chia sẻ quyền Người đóng góp cho SubFolder1 với contributor1.

Kết quả mong đợi: contributor1 có thể thêm, chỉnh sửa và xóa File trong SubFolder1.

Kịch bản 2: Người đóng góp thêm và chỉnh sửa File

User: contributor1 có quyền Người đóng góp trên Specifications.

Hành động: Thêm Spec3.pdf, đổi tên Spec1.pdf thành Spec1\_Final.pdf.

Kết quả mong đợi: File được thêm và đổi tên thành công.

Kịch bản 3: Người đọc cố gắng chỉnh sửa File

User: reader1 có quyền Người đọc trên File3.docx.

Hành động: Cố gắng xóa File3.docx.

Kết quả mong đợi: Hành động thất bại với lỗi từ chối quyền truy cập.

Các bảng và kịch bản này giúp minh họa cách User, Drive, Folder, File và quyền truy cập tương tác trong hệ thống.