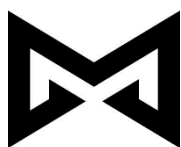


DEVICE INTEGRATION ENGINEER

ENTRANCE TEST

Người thực hiện: Nguyễn Vũ Quang

Người hướng dẫn: Hiếu Trần



Device Integration Engineer

Device Integration Team

Fossil Vietnam

MỤC LỤC

- Các nội dung chính 3
 - 1 Tổng quan..... 4
 - 2 Mô tả bài toán..... 5
 - 3 Giải pháp 6
 - 4 Cấu trúc của Project 8
 - 5 Cấu trúc ứng dụng..... 10
 - 6 Hướng dẫn sử dụng 12
 - QuickAccess 12
 - External Storage 13
 - Recycle Bin..... 15

Files System Management

Các nội dung chính

- Giải pháp giải quyết yêu cầu bài toán
- Thời gian làm bài, đường dẫn đến project (Github)
- Cấu trúc của ứng dụng
- Cấu trúc của Project
- Hướng dẫn sử dụng

1

Tổng quan

- Thời gian lập trình: 26 giờ
- Thời gian làm báo cáo: 2 giờ
- Đường dẫn đến git: <https://github.com/vuquangkhtn/FileManager>

2

Mô tả bài toán

- Hiển thị danh sách tệp tin và thư mục trong Bộ nhớ ngoài (External Storage) của thiết bị
- “Quick Access”: hiển thị danh sách tệp tin được mở (bởi ứng dụng) trong 2 ngày trở lại
- Truy cập vào thư mục
- Sao chép / Di chuyển / Xóa / Tạo thư mục / Tập tin
- Có tính năng Undo / Redo
- Hiển thị thông tin của tập tin / thư mục gồm:
 - o Ngày tạo
 - o Lần thay đổi cuối
 - o Lần truy cập cuối (Bởi ứng dụng)
 - o Kích thước (số lượng tập tin và thư mục con đối với thư mục, bộ nhớ sử dụng đối với tập tin)
- Sắp xếp Tập tin / Thư mục theo:
 - o Tên
 - o Thời gian (Tạo, Thay đổi, Truy cập)
 - o Loại tập tin / thư mục
- Lưu log tất cả hành vi người dùng (truy cập, tạo, sao chép, di chuyển, xóa , sắp xếp). Có thể cấu hình để lựa chọn in ra console hay in ra file

Môi trường hoạt động dự kiến: Android (4.1.1 – 8.0)

Công cụ phát triển dự kiến: Android Studio.

Ngôn ngữ phát triển: Java, Android SDK, JDK

3

Giải pháp

STT	Yêu cầu	Giải pháp
1	Hiển thị danh sách tệp tin và thư mục	Lấy danh sách toàn bộ File do hệ thống trả về theo đường dẫn External Directory. Phân loại tệp tin và thư mục để hiển thị biểu tượng và thông tin tương ứng. Các thông tin của tệp tin gồm tên, kích thước, loại tệp tin, lần thay đổi cuối, ngày khởi tạo đều được hệ thống trả về. Riêng thời gian truy cập sẽ được quản lý dưới local database
2	Quick Access	Với mỗi tệp tin người dùng vừa truy cập sẽ lưu lại thời gian truy cập và đường dẫn của tệp tin đó xuống local database. Khi người dùng truy vấn sẽ lấy toàn bộ tệp tin dưới database và kiểm tra tệp tin nào có thời gian truy cập trong 2 ngày trở lại và hiển thị ra cho người dùng
3	Truy cập vào thư mục	Tương tự như phần 1 nhưng thay đường dẫn đến External Directory bằng đường dẫn của thư mục muốn truy cập
4	Sao chép / Di chuyển / Xóa / Tạo thư mục / Tệp tin	Viết hàm cho phép sao chép / xóa / di chuyển tệp tin hoặc thư mục (bao gồm tất cả các tệp tin và thư mục con trong thư mục đó).
5	Undo / Redo	Sử dụng Command Pattern: <ul style="list-style-type: none"> - Tạo class quản lý Action tên là ActionManager có 2 Stack để lưu trữ các hành động đã được thực hiện và các hành động bị Undo - Với mỗi hành động (sao chép / xóa / di chuyển) được thực hiện, ta tiến hành lưu lại hành động đó vào ngăn xếp (Stack)

		<ul style="list-style-type: none"> - Khi người dùng Undo, lấy hành động đầu Stack ra thực hiện hàm undo, sau đó lưu hoạt động đó qua một ngăn xếp khác (UndoStack) - Khi người dùng Redo, lấy hành động trong UndoStack ra để thực hiện và chuyển qua lại cho Stack - Khi người dùng thực hiện hành động mới, xóa các hành động được lưu trong UndoStack
6	Hiển thị thông tin của tập tin / thư mục	<p>Ngày tạo: Hệ thống trả về (Chỉ đúng cho API 26 trở lên)</p> <p>Lần truy cập cuối: Thời gian truy cập cuối của tập tin được lưu trữ dưới database (Như đã đề cập trong Yêu cầu 2)</p> <p>Các thông tin còn lại có thể lấy được từ hệ thống trả về</p>
7	Sắp xếp Tập tin / Thư mục	<p>Sau khi lấy được danh sách tập tin / thư mục, tiến hành phân loại và sắp xếp theo các lựa chọn được đề ra trước khi hiển thị danh sách ra cho người dùng.</p> <p>Ở đây, ta sử dụng Collections.sort để sắp xếp. Tương ứng với mỗi lựa chọn sẽ tạo một Comparator tương ứng cho lựa chọn đó</p>
8	Lưu log tất cả hành vi người dùng	<p>Tạo lớp Flog – là nơi quản lý in log các thao tác của người dùng trên ứng dụng. Ta có thể điều chỉnh yêu cầu in log trên Console hay File. File sẽ được lưu ngay trên đường dẫn External Storage với tên là “log_filemanager.txt”</p>

4

Cấu trúc của Project

✚ App: các class chứa thông tin toàn cục (global) hỗ trợ cho các chức năng của ứng dụng, có thể truy cập bởi mọi class khác mà không cần khởi tạo lại

✚ Base:

- Các Activity, Dialog và Fragment cơ sở, được cài đặt sẵn một số tính năng cơ bản như hiển thị loading, hiển thị tin nhắn, hiển thị lỗi, bắt sự kiện bấm Back, ấn bàn phím,... cho phép các Component con kế thừa
- Mẫu Presenter để sử dụng cho mô hình MVP

✚ Data:

- Dao: truy cập và lấy dữ liệu từ local database. Cụ thể là quản lý thông tin của các file được truy cập bởi ứng dụng.
- Manager:
- ActionManager: Quản lý các hành động được thực hiện bởi người dùng
- OpenedFileManager: cung cấp các chức năng cần thiết liên quan đến các tập tin được truy cập và lịch sử truy cập (dữ liệu được DAO trả về)
- AppDataManager: Quản lý cả ActionManager và OpenedFileManager. Sử dụng mô hình Singleton để chỉ khởi tạo một đối tượng khi mở ứng dụng và sử dụng đối tượng đó xuyên suốt quá trình hoạt động của ứng dụng
- Model: Chức các cấu trúc dữ liệu
 - Action: Dữ liệu về các hành động của người dùng (Tạo/xóa/di chuyển/copy)
 - File: dữ liệu về File, tương tự như File do hệ thống trả về nhưng tùy biến một số thông tin như thời gian truy cập để hỗ trợ cho các chức năng của ứng dụng

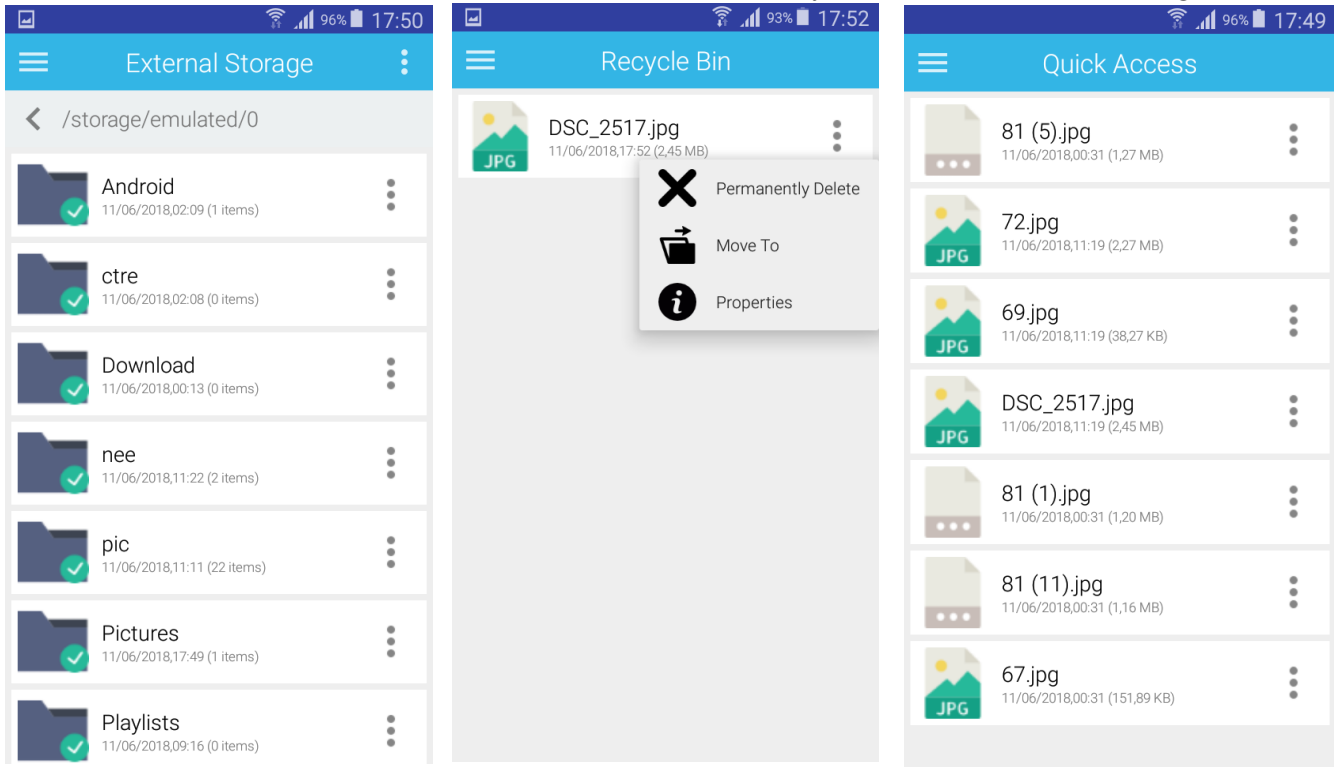
- ✚ Dialog: Chứa custom dialog. Cụ thể có dialog cung cấp chức năng cho phép người dùng chọn đường dẫn và trả về đường dẫn mà người dùng chọn.
- ✚ Main: chứa cài đặt và giao diện màn hình chính của ứng dụng
 - MainActivity: Activity chính của ứng dụng
 - MainPresenter: Xử lý các event do activity trả về, kết nối với AppDataManager để lấy các thông tin cần thiết. Là nơi trao đổi thông tin giữa tầng data và view.
- ✚ Utils: chứa các lớp (chỉ có các phương thức tĩnh) thực hiện các thao tác trên một loại đối tượng, không cần khởi tạo

- ▶ app
- ▶ base
- ▼ data
 - ▶ dao
 - ▶ manager
 - ▼ model
 - ▶ action
 - ▶ file
- ▶ dialog
- ▶ main
- ▶ utils

5

Cấu trúc ứng dụng

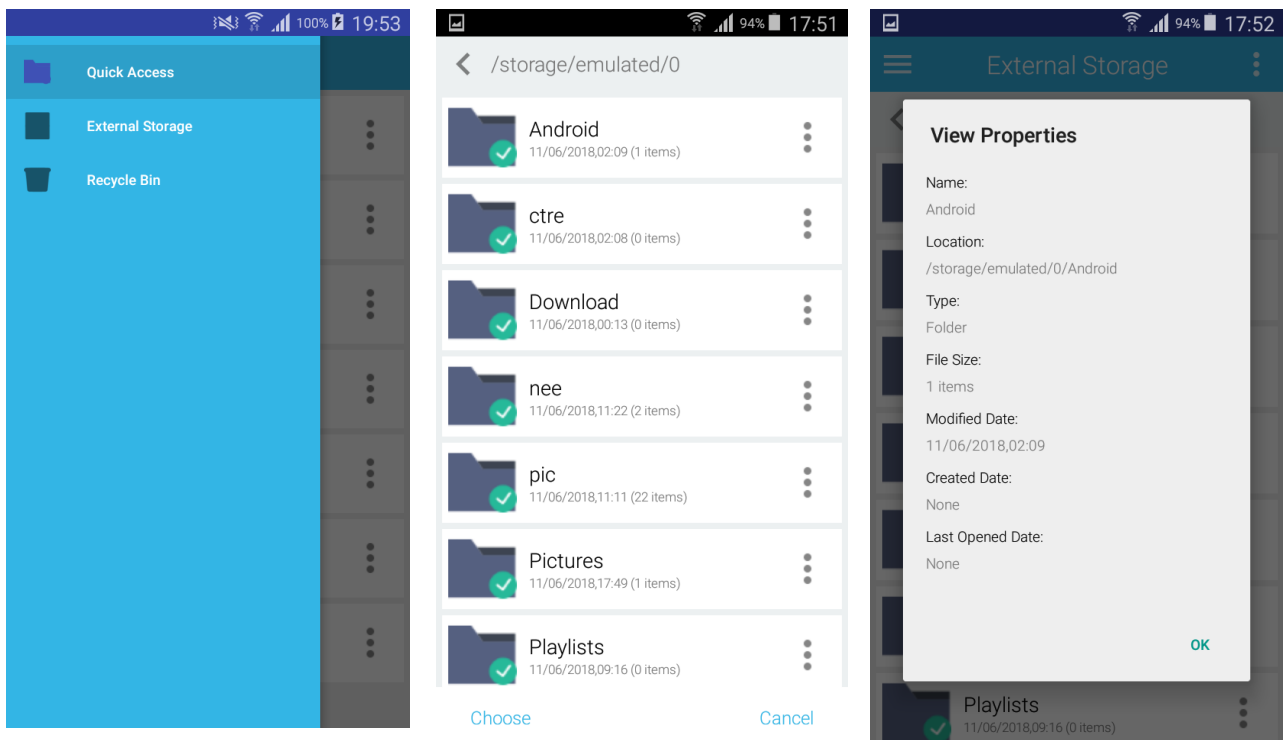
- Màn hình chính: Gồm màn hình QuickAccess, Recycle Bin và External Storage



- Mỗi tập tin hiển thị popup cho phép người dùng mở thư mục, xóa, sao chép, di chuyển, xem thông tin tập tin



- Hiện thị popup trên action bar cho phép chọn nhiều files, sắp xếp các files theo tên, thời gian, loại file, cho phép tạo file, tạo folder, cho phép thực hiện chức năng undo, redo.
- Navigation cho phép chuyển đổi các màn hình
- Dialog cho phép chọn đường dẫn
- Dialog xem thông tin chi tiết của tập tin
- Dialog chọn tùy chọn để sắp xếp file
- Dialog nhập tên file/thư mục muốn tạo
- Màn hình hiển thị khi danh sách tập tin rỗng
- Màn hình để chọn multifiles và thanh chức năng có các chức năng delete, copy, move

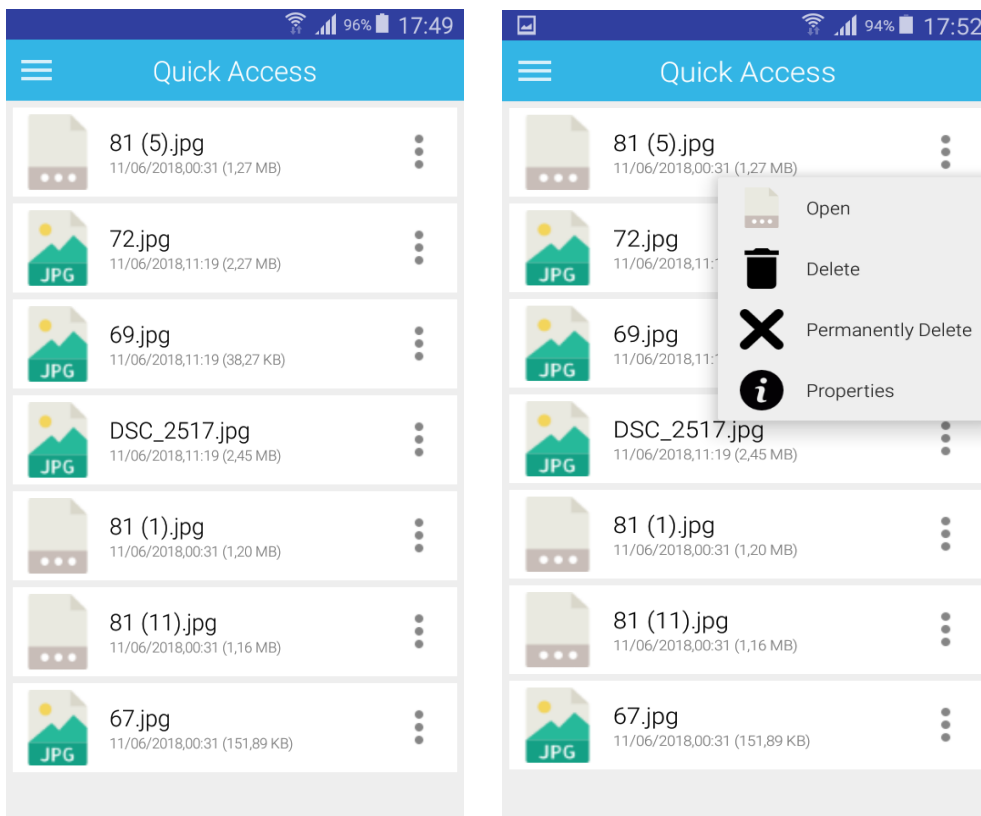


6

Hướng dẫn sử dụng

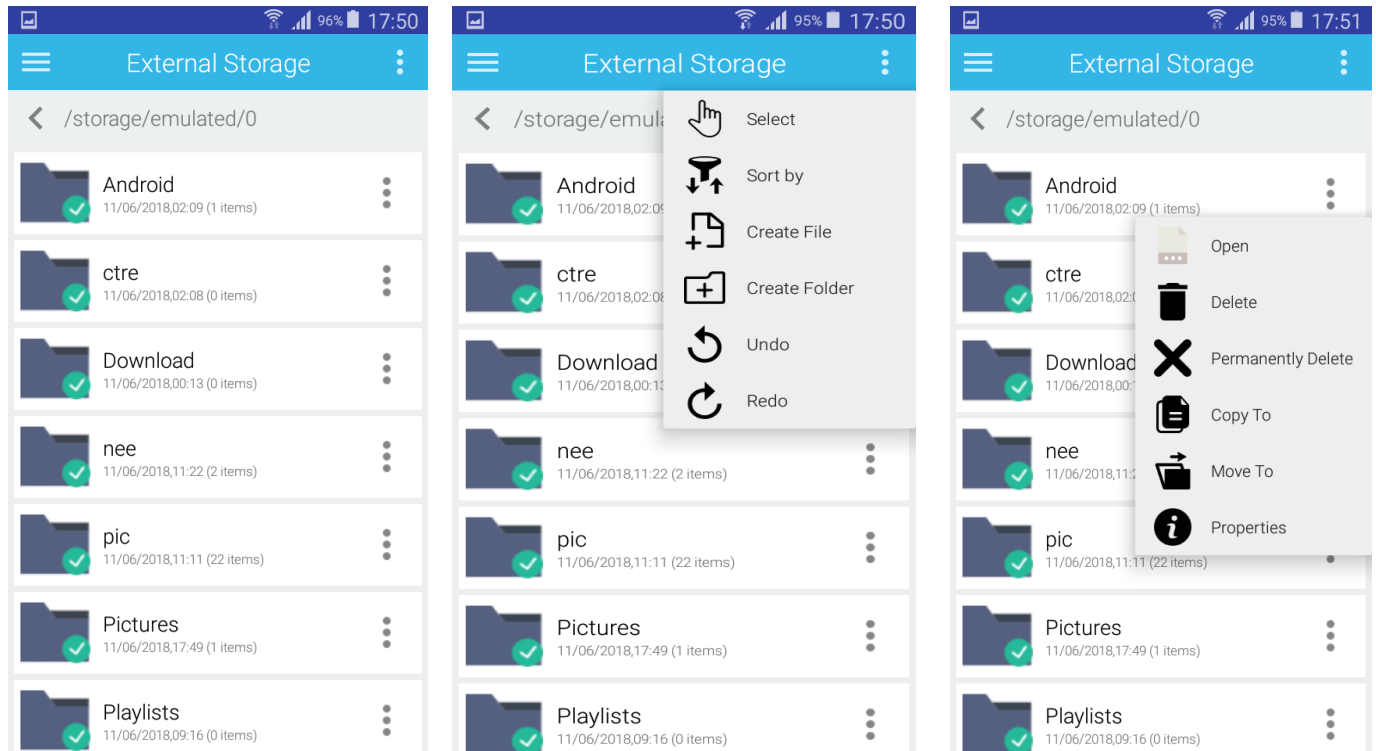
QuickAccess

- Màn hình hiển thị đầu tiên lúc khởi động ứng dụng
- Hiển thị thông tin các tập tin được truy cập gần đây
- Cho phép thao tác một số chức năng trên file gồm mở file, xóa file, xóa hoàn toàn file, xem thông tin file



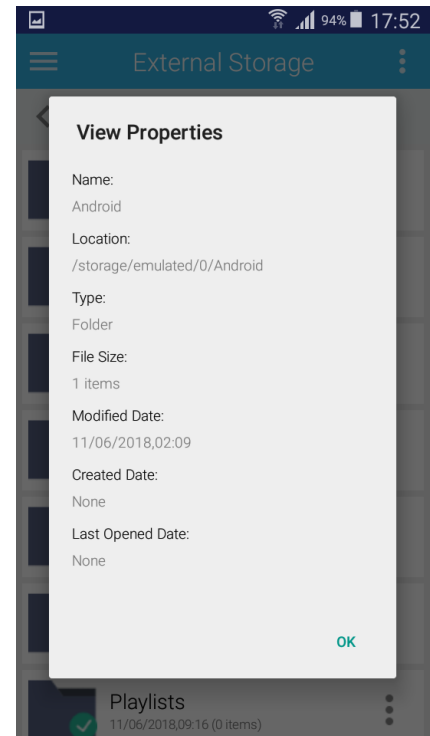
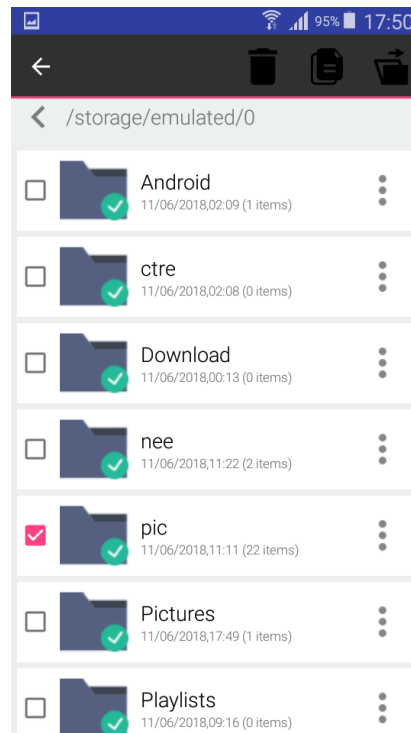
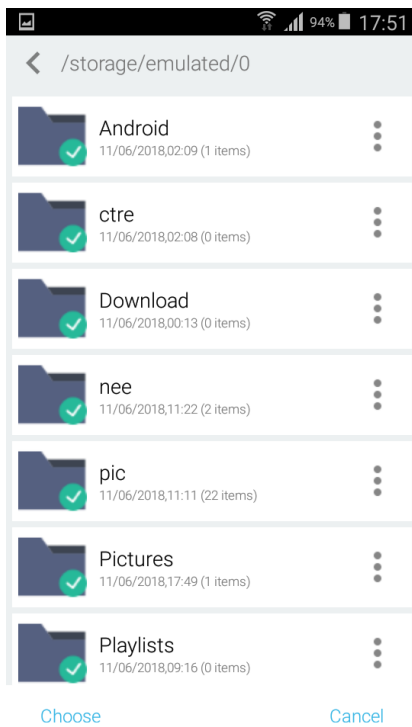
External Storage

- Màn hình hiển thị danh sách các tập tin và thư mục của hệ thống
- Cho phép khởi tạo / xóa / sao chép / di chuyển một hoặc nhiều tập tin / thư mục
- Có thể undo/redo các hành vi người dùng



- Khi chọn “Copy”, “Move”, hiển thị dialog chọn đường dẫn đến thư mục muốn copy/move đến
- Bấm “Choose” để chọn thư mục đích, bấm cancel để hủy
- Chọn “Select” để hiển thị giao diện chọn nhiều tập tin
- Chọn “Properties” để xem thông tin tập tin
- Hiển thị màn hình nhập tên file khi chọn “Create File”, tạo thêm file tại thư mục hiện tại nếu tạo thành công

- Hiển thị màn hình nhập tên folder khi chọn “Create Folder”, tạo thêm folder tại thư mục hiện tại nếu tạo thành công



Recycle Bin

- Màn hình hiển thị danh sách các tập tin và thư mục bị xóa (chưa xóa hoàn toàn)
- Cho phép xóa hoàn toàn tập tin, di chuyển tập tin ra khỏi Recycle bin, xem thông tin của tập tin

