File Manager Project

1 Objective

In this project, we want you to create a Java application with GUI interface. You will mainly uses Swing components for creating a user interface (GUI) and then implementing the functions of the application using java programming. This project entitled as (File Manager) and this project assigned for this course.

2 Description

2.1 User interface (GUI)

The user interface of the program contains 1 main JFrame:
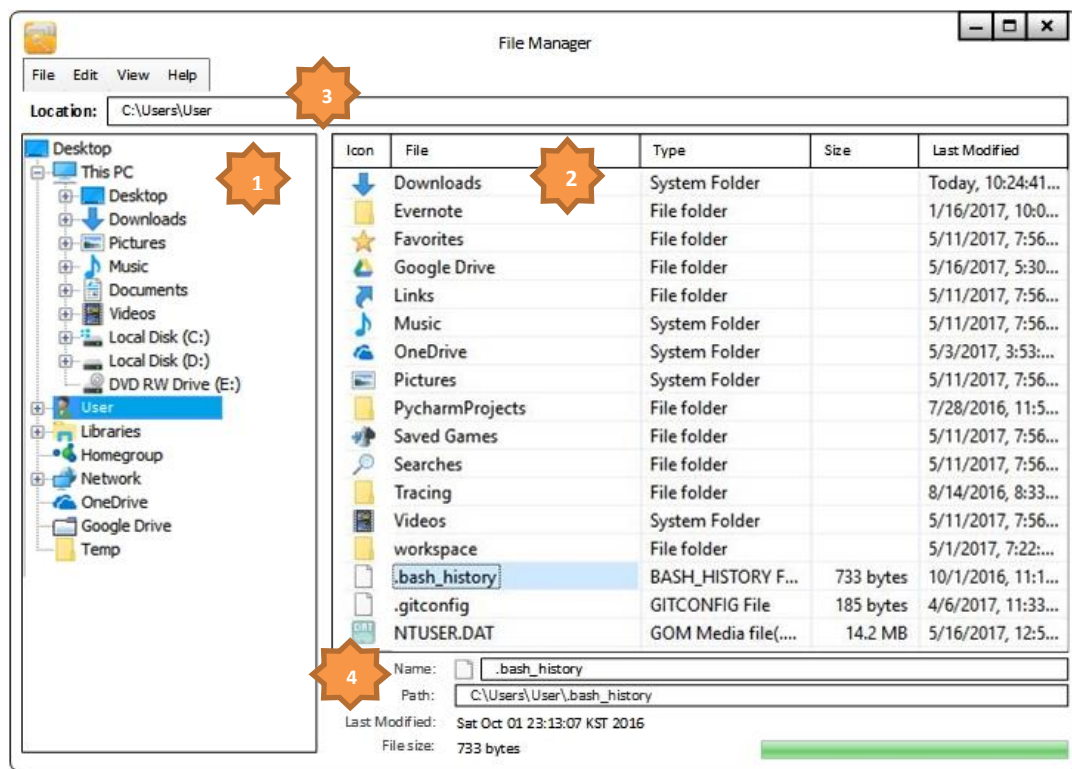
- File Manager

Figure 1: File Manager GUI

A suggested GUI of the File Manager has been illustrated in Figure 1. *You can freely design your own GUI if you want but you must include all the required functions of the project in your GUI design.*

2.2 Functions:

❖ The main functions of File Manager are as follows:
 ➢ Display system structure:
    • Display system folder structure by tree component with default its own icons (Tree component can expand and shrink) (1)
    • By clicking, it can open that folder and update display view to show files/folders in that folder
      • Update location of opened folder (3)
      • By double-clicking, tree view will expand and show inner folders
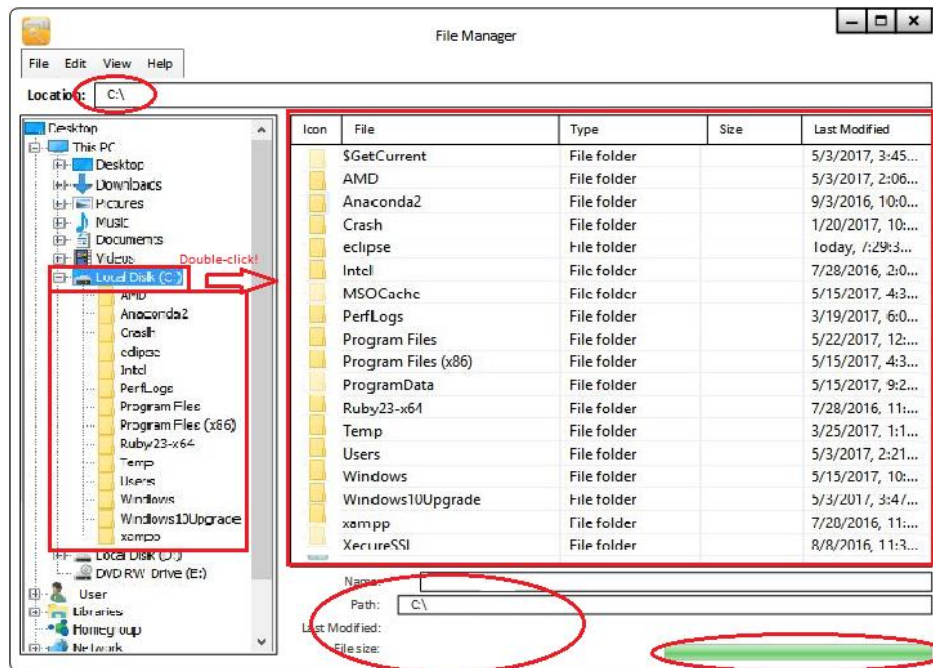
    • Show opened folder's files/folders on display view (table/list/icon) (2)
      • Show opening action progress in progress bar
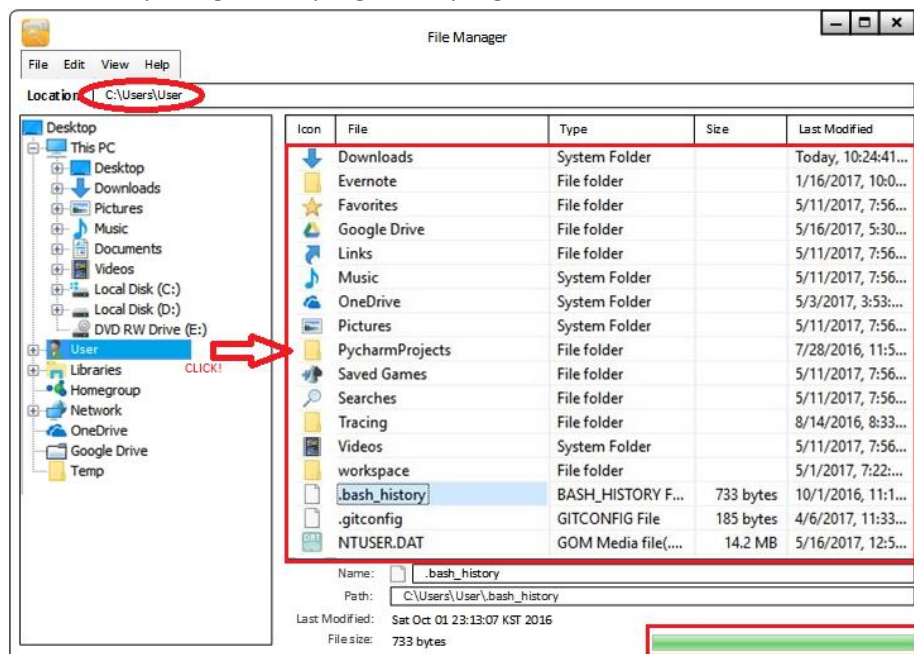
- Show selected file/folder's information in information section (panel) (4)
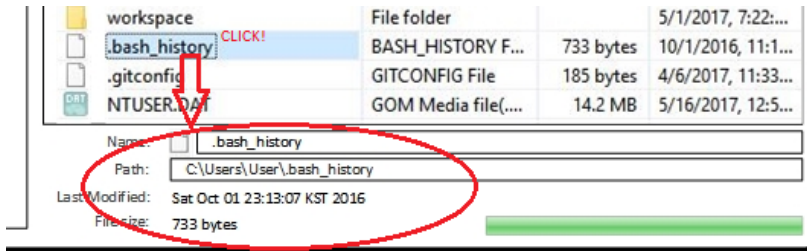
- If user double-click on folder, open that folder and update GUI as like above
- If user double-click on file, open file with system's default application
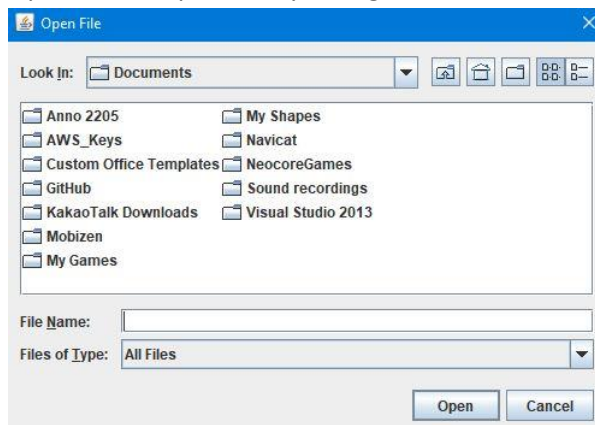- User can enter location address to open and update GUI by location address bar:

- Menu:
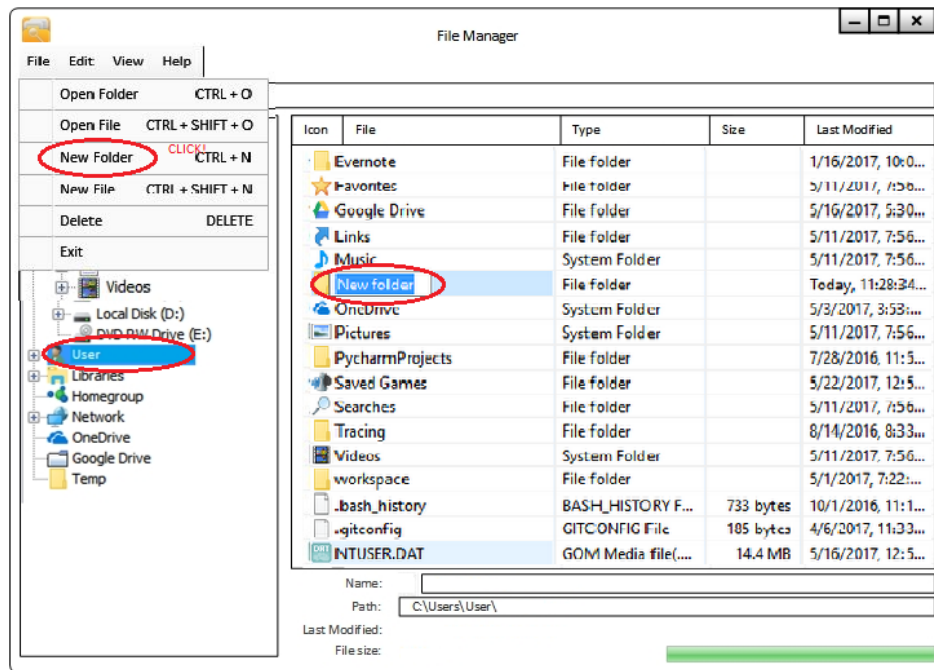 - File
    ♦ Open Folder -> Open folder by dialog box

    ♦ Open File -> Open file by dialog box
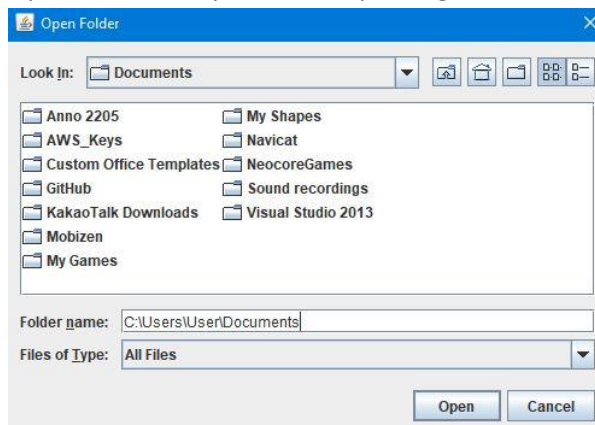
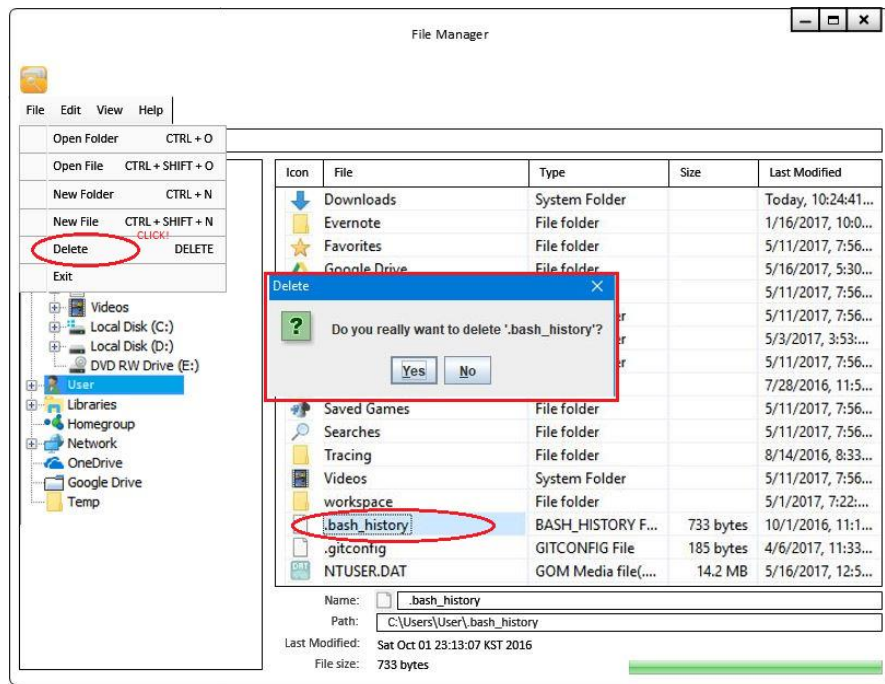| File		
Open Folder	CTRL + O
Open File	CTRL + SHIFT + O
New Folder	CTRL + N
New File	CTRL + SHIFT + N
Delete	DELETE
Exit	

♦ New Folder -> Create new folder on opened location

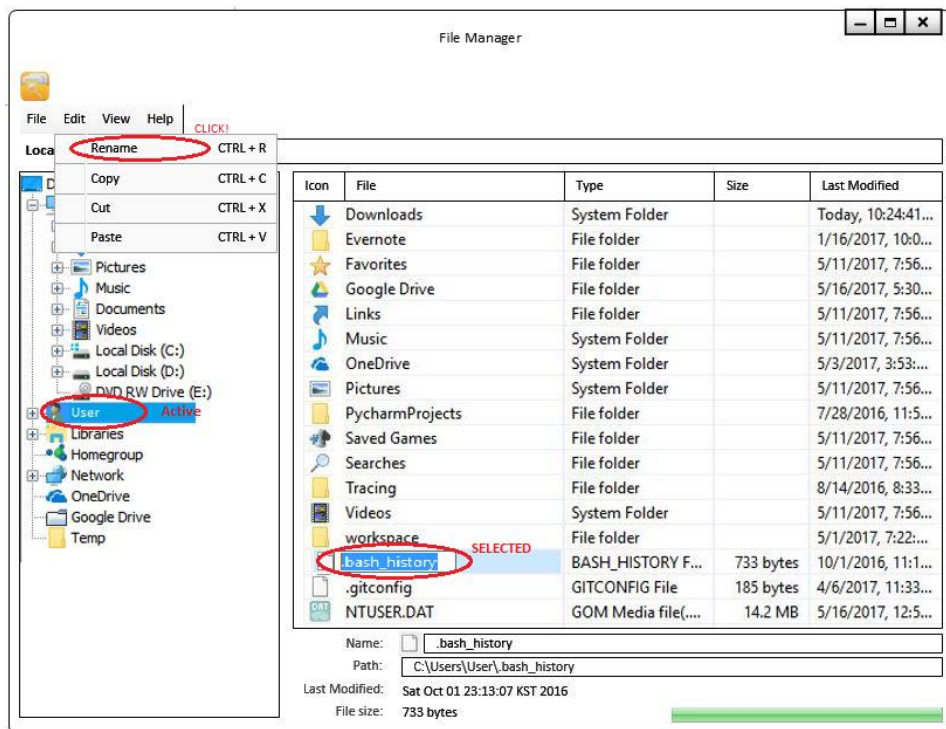♦ New File -> Create new text file on opened location
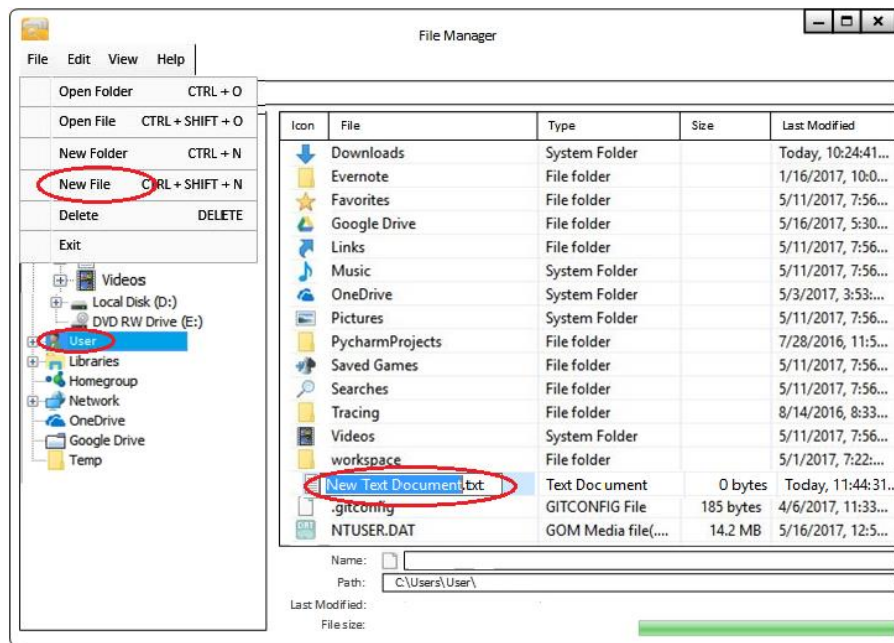
♦ Delete -> Delete file/folder that selected in display view
 ➢ Show confirm dialog to delete or not

♦ Exit -> Close application

• Edit

  ♦ Rename -> Rename selected folder/file's name

- ◆ Copy -> Copy selected file/folder (Save file/folder's path to paste)
- ◆ Cut -> Move selected file/folder (Similar with copy but after paste, delete old file/folder)
- ◆ Paste -> Paste copied file on opened folder (location)

- • View (Similar with OS file manager application)
  - ◆ Details -> (Default) Show files/folders detailed view with information
  - ◆ List -> Show files/folders in listed view (just name and icons)
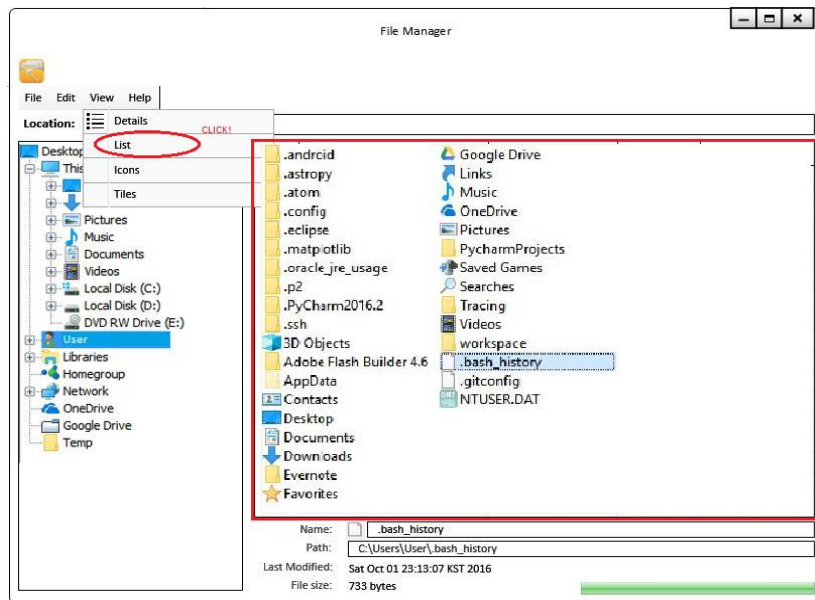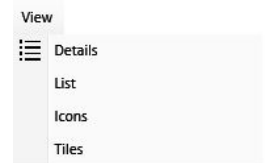
- ◆ Icons -> Show files/folders with medium icons on display view (Extra)
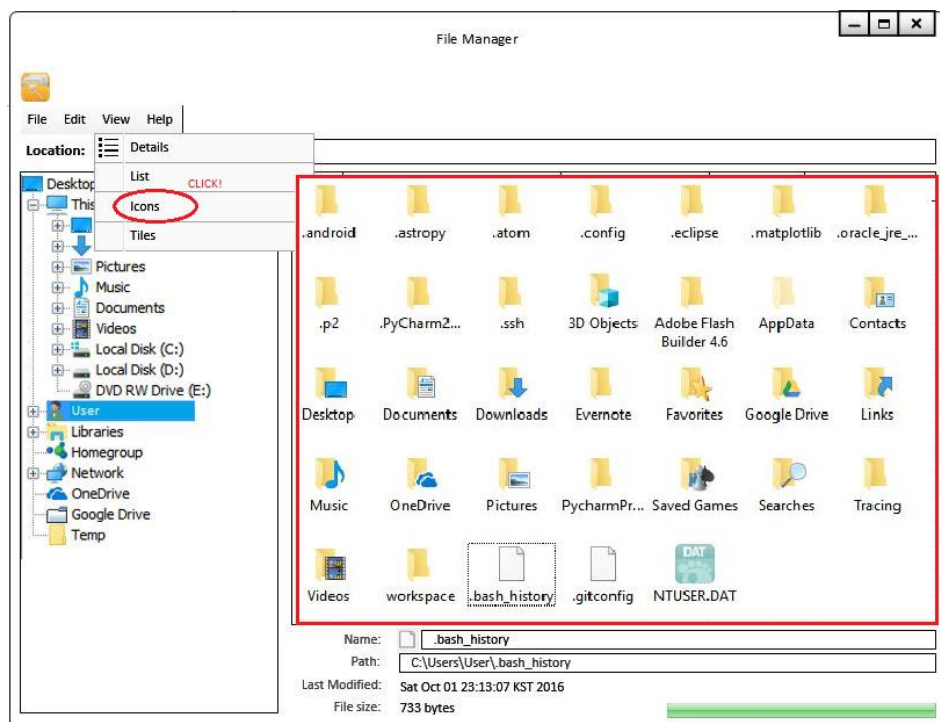
♦ Tiles -> Show files/folders with icons and type (Extra)
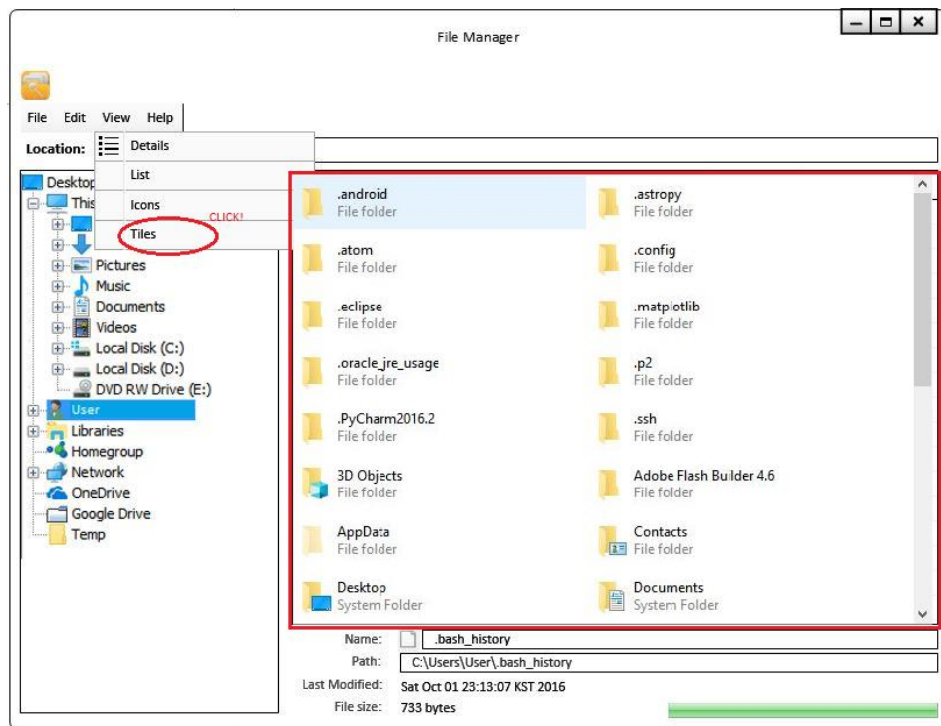
- Help -> It shows list of keyboard shortcuts with description (For example: CTRL + C = Copy)

➢ Change view mode:
- Change view of file/folder display based in selected mode from menu bar. Details is default mode, must have List mode; Icons and Tiles mode is extra.
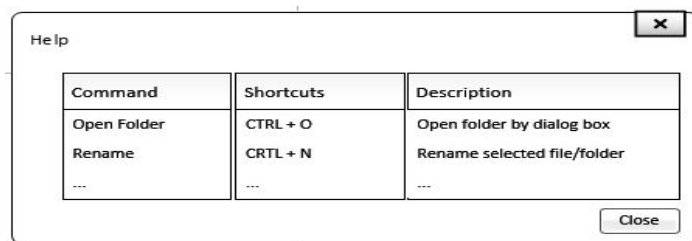- Hint: It may need JTable (Details), JList components, icon render functions, cardLayout or other things. It's totally up to you.

➢ Handle keyboard shortcuts:
- User can use keyboard shortcuts (At least simple shortcuts)
- Shortcuts: CTRL + c, CTRL + x, CTRL + v, Delete, Enter
- If you add more shortcuts, it will be extra
- Shortcut's description must be in help menu
- Hint: To get the keyboard user type, you can use KeyboardFocusManager

➢ Every folder opening action must use thread for progress bar
- When user opens folder, progress bar should show progress of opening action and update GUI simultaneously

❖ Use file:
  ➢ Save last location of folder and when it's open again it should open last location
  ➢ Also, save view mode

3 Extra Credit

In this assignment, you can earn extra credit by

  ➢ Designing friendlier user interface (better than the suggested GUI).
  ➢ Adding more functions to the program such as pop-up context menu (right-click menu)
  ➢ Add more shortcuts
  ➢ Add more view modes
  ➢ Add more menus and actions
  ➢ Add tab controller on display view to open more folders

4 Project Submission

Here are the details for how to submit your project:

1. Create an executable JAR file named **FileManager.jar** that contains your runnable application.
2. Create a new directory using your student IDs. Example: 20170001_20170002
3. Put the jar file into the directory, and also your Eclipse project folder.
4. Put also **README.txt** file to the directory. This file should contain student name (in English), the version of Java you used as well as any special info we might need to know about your program (example: let us know if you did some extra things).
5. ZIP this directory and submit it via e-class.

Notes:

All the comments on the code and the names of variables, classes must be in English.

Please take care to remove any platform dependencies, such as hardcoded path names or dependence on a particular look-and-feel that may not exist on all platforms. Also, if you use any images in your application, please make sure that you included these images in your JAR file and that your code will refer to them and load them properly when they're in this JAR file (see this page for some details on how to include and load images from within a JAR file).

5 Grade

Grading for this assignment will follow this breakdown:

  ✓ 50% functionality
  ✓ 30% architectural design, coding style (code convention)
  ✓ 20% commenting
  ✓ 20% extra credit

Please let the LA or the professor know if you have any questions or something you don't understand by visiting Lab # 307 at hi-tech Center or by asking question on e-class.

If you have questions about the project requirement, I suggest you post it in public so other student can see them too but if you have a question about your project implementation, then post the question in private.

Good luck!