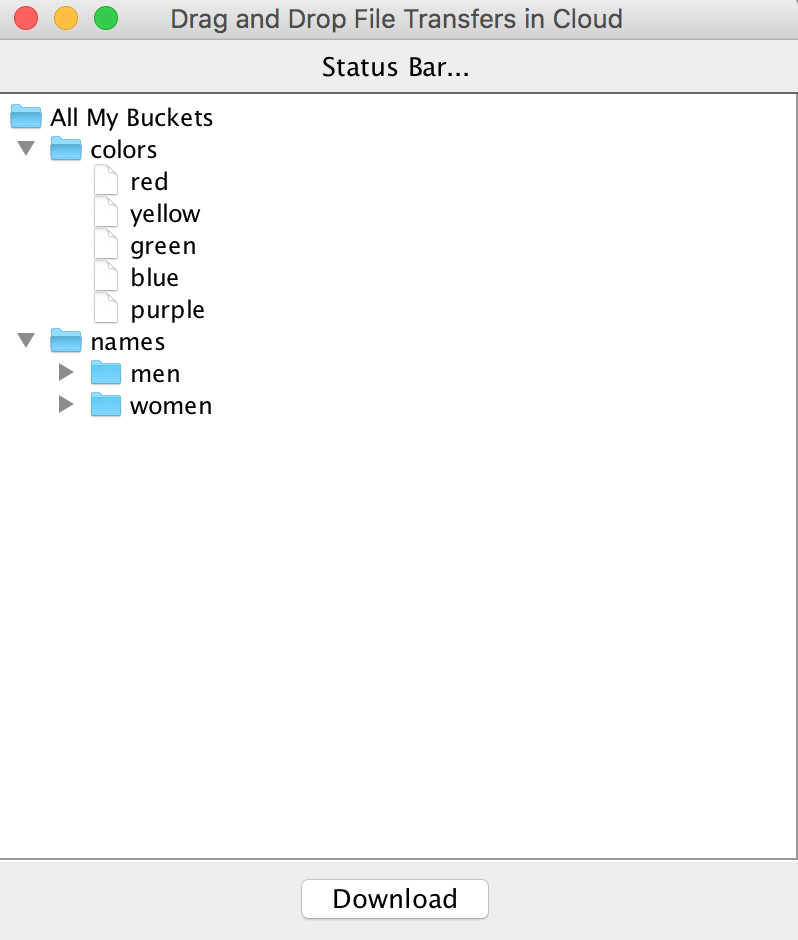
**CSCD467/567 HW Drop Box-Like Cloud Application**

**To turn in: please wrap up all your java source files into a zip file, then** turn in the single zip file on the **EWU Canvas** by going to CSCD467-01 course page on Canvas, then clicking Assignments🡪HW6->submit. Please name your zip file with your last name, followed by the initial of your first name, followed by hw6. For example, if you are John Smith, name you file as smithjhw6.zip

**What has been provided**

A Java GUI program has been included in the start-up package of this homework, with GUI buttons, JTree and other components having been set up already. **To use the provided java** file in Eclipse, you can first create an empty AWS Java Project (say HW6), as we did in lab6. Then, right click the item **src/main/java** under your Project name HW6, click **Import**, click “**General**” tab, choose “**File System**”, click **next, then “Browse”.** Then, you can navigate to the downloaded Hw6 package and choose/open the provided DragDropFiles.java file.

When running this program, you will see the GUI window, as shown in below.

****

On the top region, right below the window title, a Status Bar is designed to display a message to users after each successful uploading and downloading operations. In the bottom of the window is a ‘Download’ button. In the middle section of the window, using a JTree GUI component, it displays a file-system tree that consists of files, directories and sub-directories. Even though we have discussed the program in classroom, you are encouraged to study the program intensively and understand it completely before you proceed to implement the homework. The provided program has already implemented the following features.

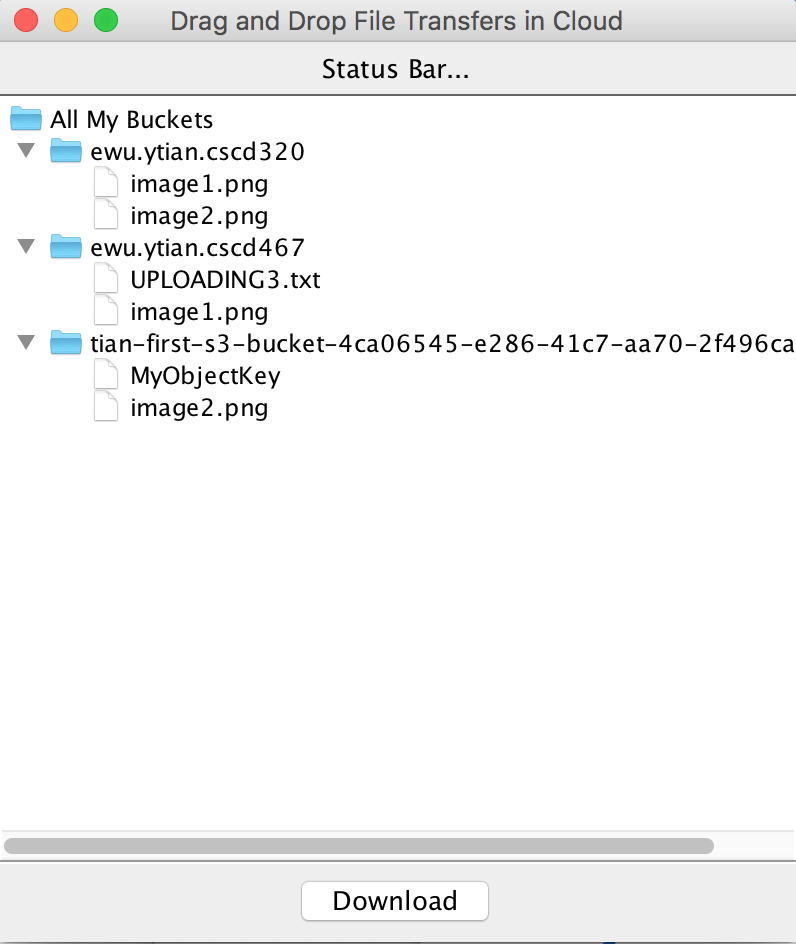
1. If you drag and drop a **text** file **F2 located on your desktop** into the ‘colors’ folders in the GUI window, the program duplicates **F2** into the same directory that holds your Java source files. Then a new entry is added to the file tree under ‘colors’ folder in the GUI. At last, the Status Bar displays a message – “Uploaded \*\* F2 \*\* successfully!”.
2. After you select an entry in the file tree in the GUI window, you click the ‘Download’ button, you will see a message pop-up that simulates a file downloading.

**Requirements to do for this homework**

Based on the provided GUI program, you are required to implement an application that resembles the drop-box application. You have the **freedom** to add, delete and update everything the provided start-up java program when trying to fulfil the requirements as listed below. As prerequisites, you have to first set up Amazon cloud account, Java AWS SDK and your AWS user ID and access key before you are able to communicate to the Amazon cloud, as we have done in lab6.

You will have to implement the following features.

1. After your program starts up, it shows a GUI window that displays a list of S3 bucket names and file names under each S3 bucket, while the bucket and the file contents are actually stored in the Amazon S3 cloud storage. We assume that these S3 buckets and files have been existing in the Amazon S3 cloud storage before you run your program. For an example,

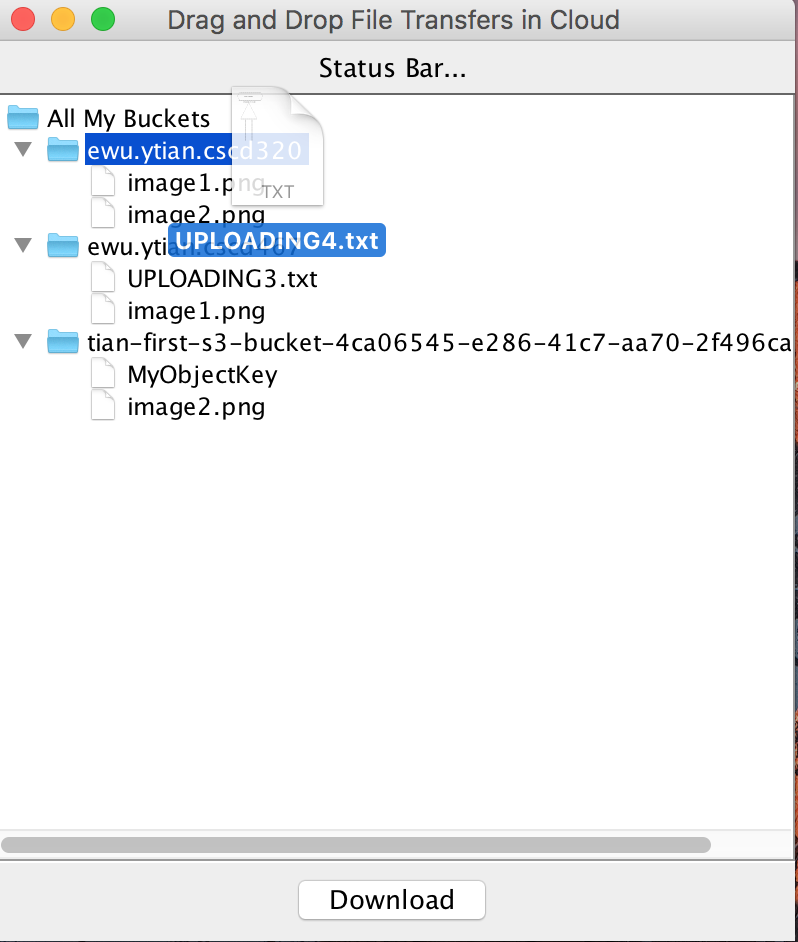


In the screenshot above, that means I actually have three buckets in the Amazon S3 cloud storage: ‘ewu.ytian.cscd320’, ‘ewu.ytian.cscd467’ and ‘tian-first-s3-bucket-…….96ca’. I have two files in the “ewu.ytian.cscd467’ bucket, UPLOADING3.txt and image1.png.

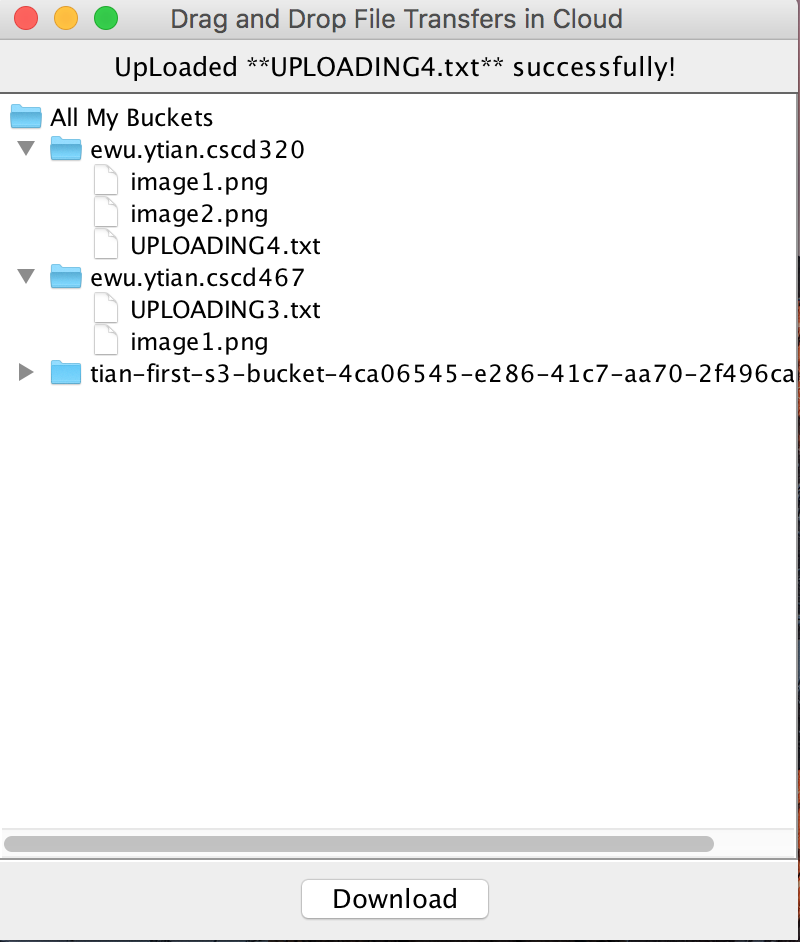
**Your program is required to display a similar set of bucket names and file names, while the buckets and the file contents are actually stored in the Amazon S3 cloud storage. You are allowed to manually create some buckets and to upload some files into them through the Amazon AWS console, before you execute your program. In this homework, assuming we only handle two levels of the file system tree, a list of buckets as the first level and a set of files in a bucket as the second level. We will NOT complicate ourselves to handle sub-directories inside a S3 bucket, unless otherwise specified in the last section of extra credits.**

1. **Uploading Programmatically**

When you drag a file **D** from your native file system (for example, from the Desktop of your computer) and drop it on one of the bucket names **B** in the GUI window. The file **D** will be uploaded and copied into the bucket **B** which is hosted in the Amazon S3 storage. Meanwhile, a new tree node entry will be added to the GUI window to reflect the upload of file **D**. The diagram below illustrates dragging and dropping a file named “UPLOADING4.txt” onto the bucket name “ewu.ytian.cscd320”.



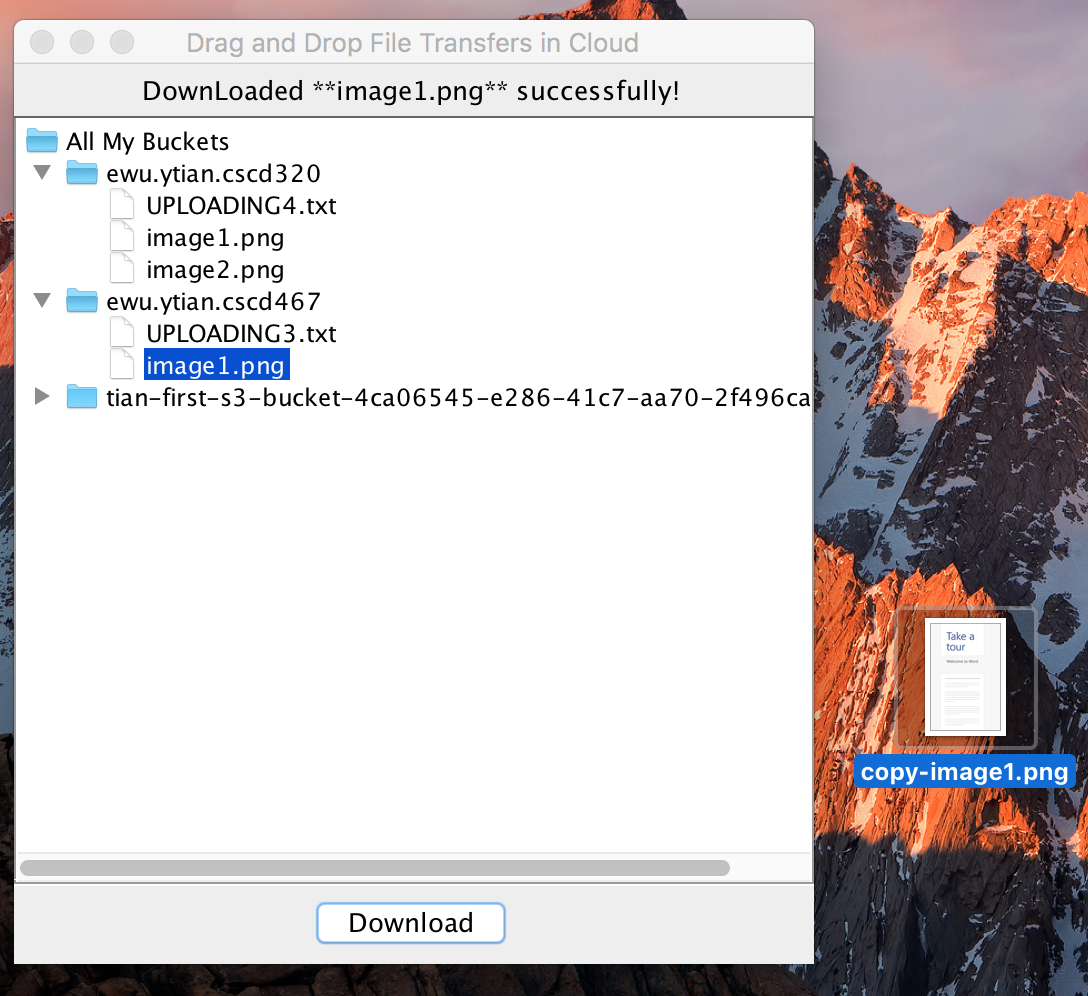
After you release your mouse click and drop the file, the file content of “UPLOADING4.txt” is literally transferred into Amazon S3 bucket “ewu.ytian.cscd320”. The following picture shows the effect **after** you drop the file.



We observed two changes in the GUI. First, on the Status Bar, a message “Uploaded \*\*UPLOADING4.txt\*\* successfully!” is displayed to indicate a successful upload. Second, a new tree node entry “UPLOADING4.txt” is added under the bucket name “ewu.ytian.cscd320” in the GUI window.

**Your implementation has to support any type of files, including binary files and text files, and to show the same behavior as depicted above.**

1. **Downloading Programmatically**



In the GUI window, after you highlight a file name by clicking it with your mouse, you can click the ‘Download’ button at the bottom of the GUI window, in order to download the file and to **copy** the file content from S3 cloud storage onto your local computer. In the picture shown above, the file **image1.png** is chosen, clicking the button ‘Download” triggers the transfer of the file from the cloud storage to the Desktop of your local computer. We can see the **copied** file **copy-image1.png** located on the Desktop of the local computer. Then a message “Downloaded \*\*image1.png\*\* successfully!” is displayed to indicate a successful download in the Status Bar. In your implementation, you can save the downloaded file into any places (though using the **current directory ./** is recommended) on your local computer, but you have to display a message on the standard output that shows where the file goes.

**Your implementation has to support any type of files, including binary files and text files, and to show the same behavior as depicted above.**

1. **Extra credits for Undergraduate students**. If you could implement another Button for Creating buckets in cloud. If that button works properly, you get 1% bonus in this homework. If you can implement a third Button for Deleting files or buckets in cloud and if works, you get another 2% bonus. **Before deleting files, please ask users for confirmation using GUI dialog box, to prevent accidental deletion**. So, the total number of bonus points is up to 3%. **Please print out a message on the standard output to notify the grader that “you have done the extra credit Part 1”. Without displaying the message can result in loss of the bonus points.**
2. Please organize your source code so that I can compile **all** your source files contained in **one** folder using command, **javac \*.java**, and run your program using command on command line, **java DragDropFiles. Please use the provided DragDropFiles.java file to house your main method.**
3. **You can have your own design for any details that have NOT been specified in this document.**
4. **Extra credits for both graduate and undergraduate students. 1) (up to 5 points) Please add a progress bar right below the status bar in the GUI, which will show the progress when uploading or downloading a file to/from the cloud. You can find reference code here about progress bar in Java,** [**https://www.geeksforgeeks.org/java-swing-jprogressbar/**](https://www.geeksforgeeks.org/java-swing-jprogressbar/)

**2)Implement the entire file system tree of all files in the cloud. In other words, your code will handle arbitrary levels of folders/subfolders in your cloud. For example, in your Bucket1, you have folder11. In folder11, you have folder 111. In folder 111, you have a file1111.txt. Your GUI window will properly display the entire hierarchy of all the files/folders in your S3 account. And your code supports deletion and creation of folder/bucket also. (up to 10 points) Please print out a message on the standard output to notify the grader that “you have done the extra credit, Part 2”. Without displaying the message can result in loss of the bonus points.**