

Report On CSE 108 Term Project

Name : Md. Mohaiminul Islam

Section : A1

Id : 1505018

. Advisor : Md Aashikur Rahman Azim

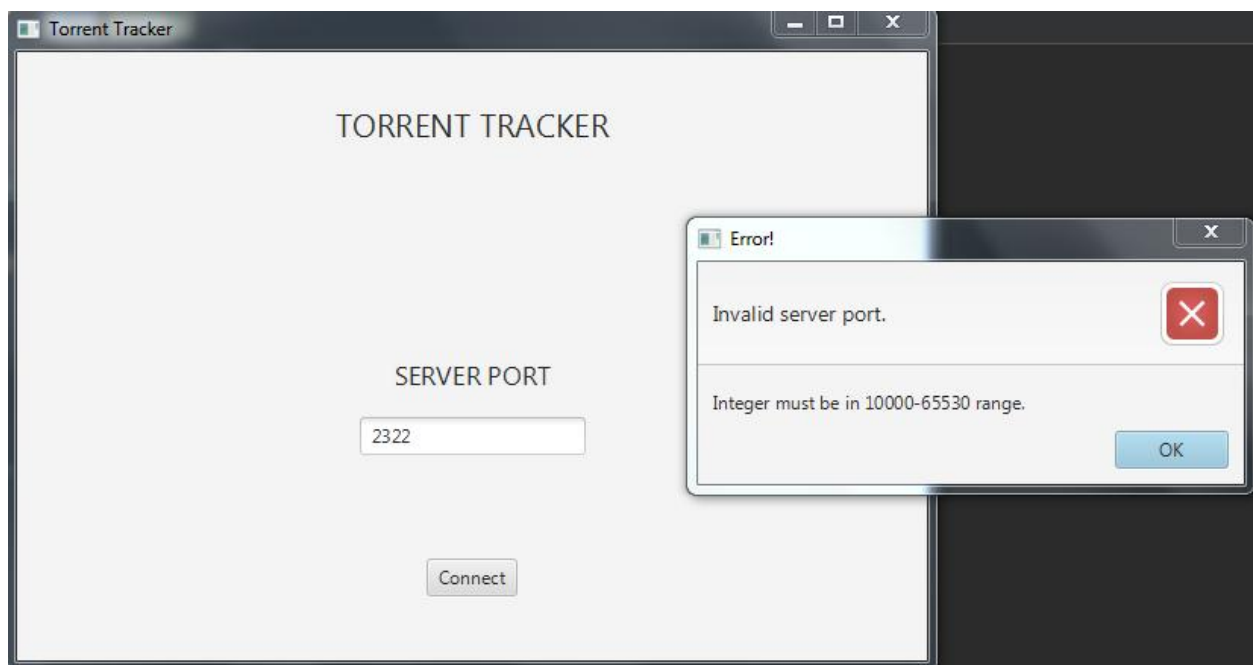
Project : Torrent Download Manager

Summary:

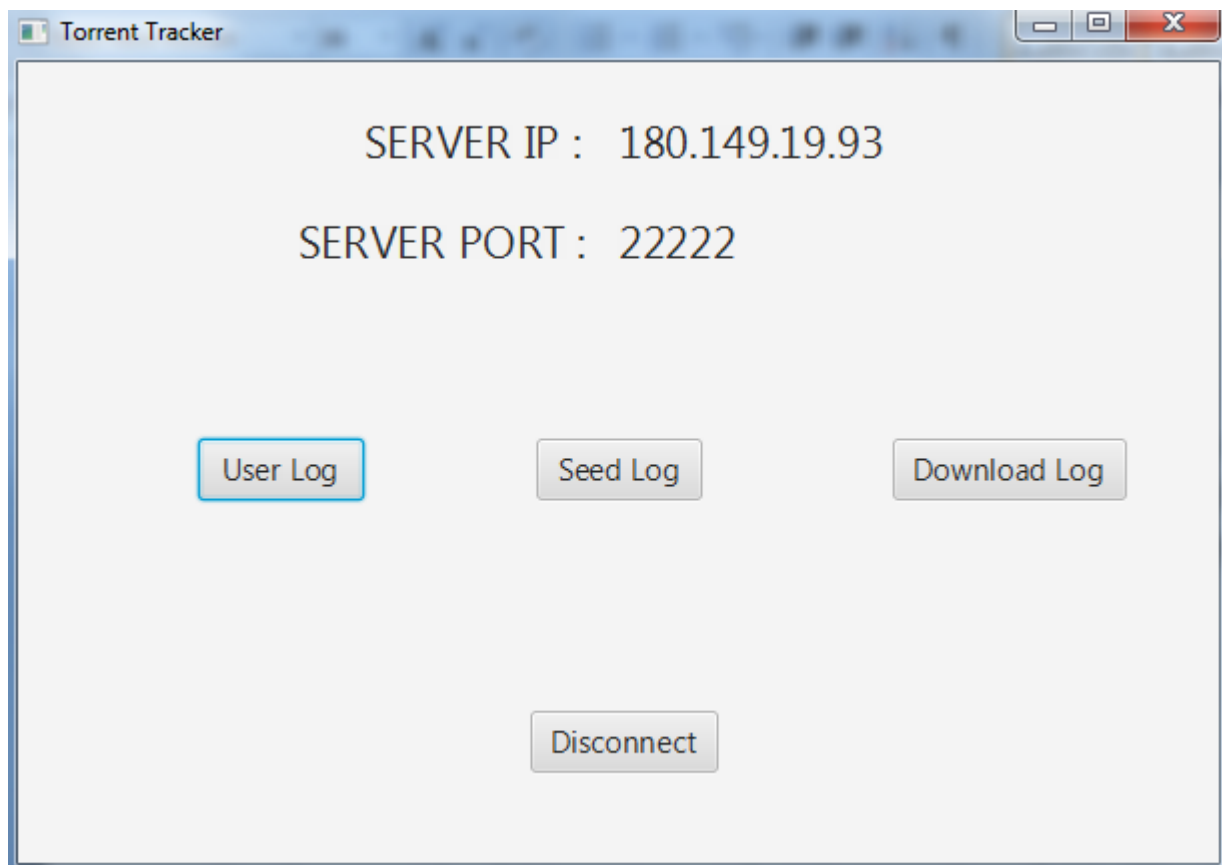
This Torrent Download Manager provides peer to peer downloading system for downloading any kind of file. There is also a tracker, which controls which users have access to which files and so on, and provides these information to the user who wants to download a particular file. The user system then connects to those other clients who have that file and downloads them in parallel . Then the user system merges them and creates the file that the user wants. Then the user can seed this file for other user to download.

Snapshots and Descriptions:

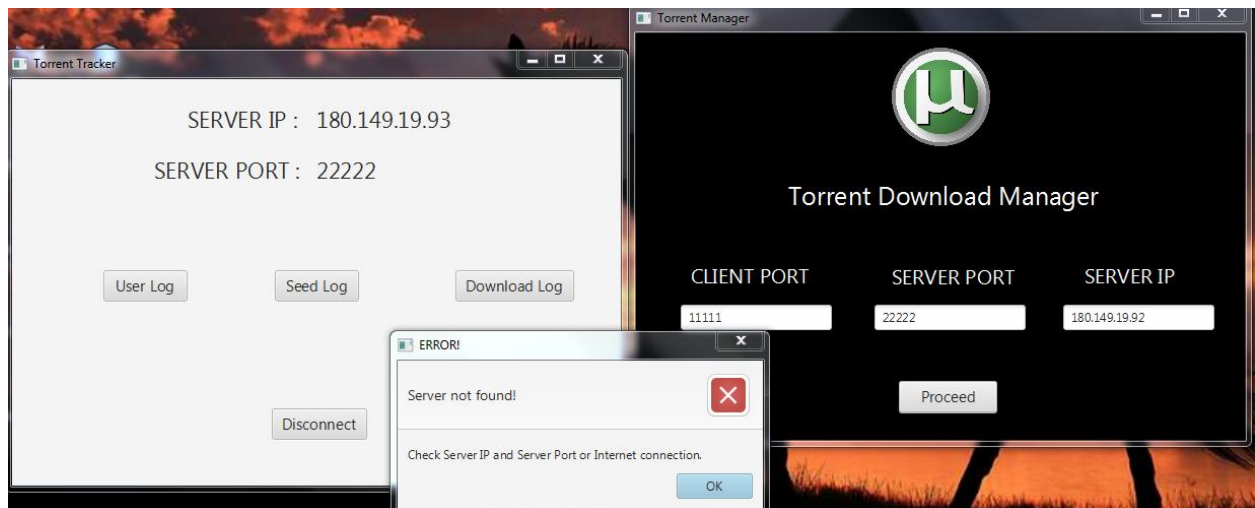
At first, the tracker must be launched at a port number within 10000 to 65530 range. Then the next page is loaded .



There are three log pages. If any user connects to or exits from this server, the user log gets updated. If someone seeds or downloads a file, the seed log or the download log gets updated respectively. Initially, all these logs are blank.

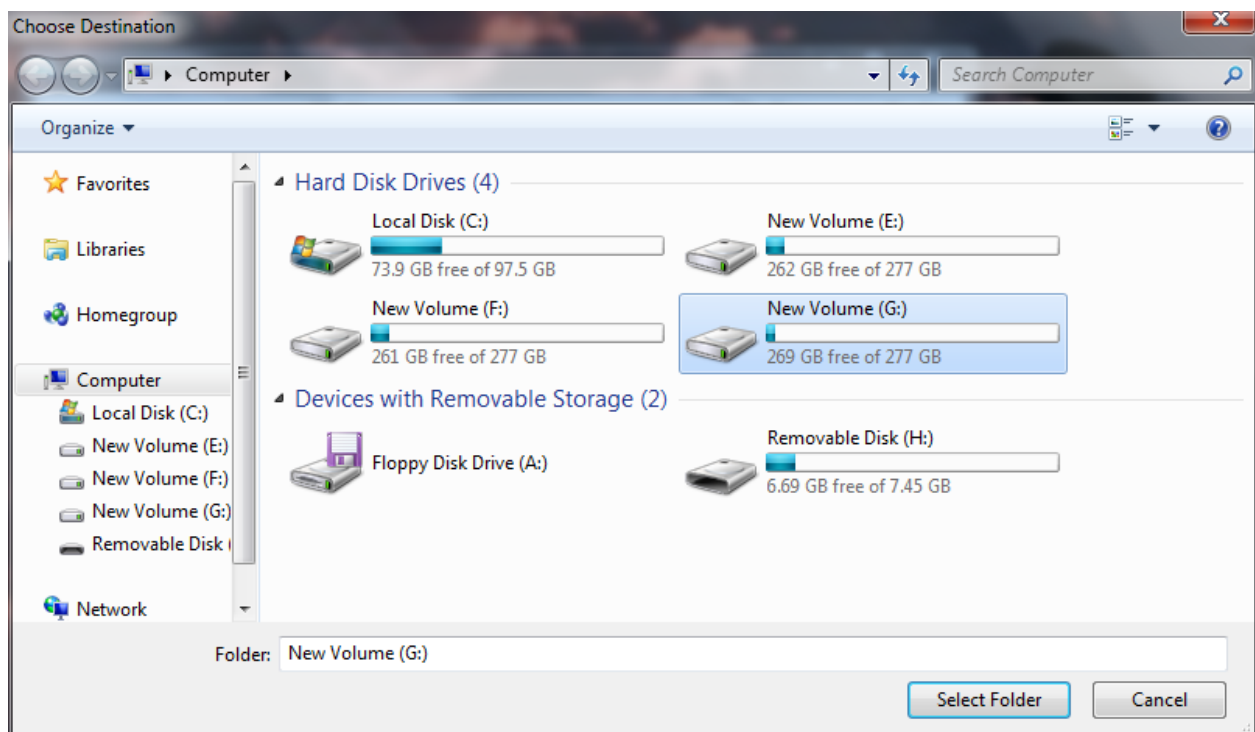
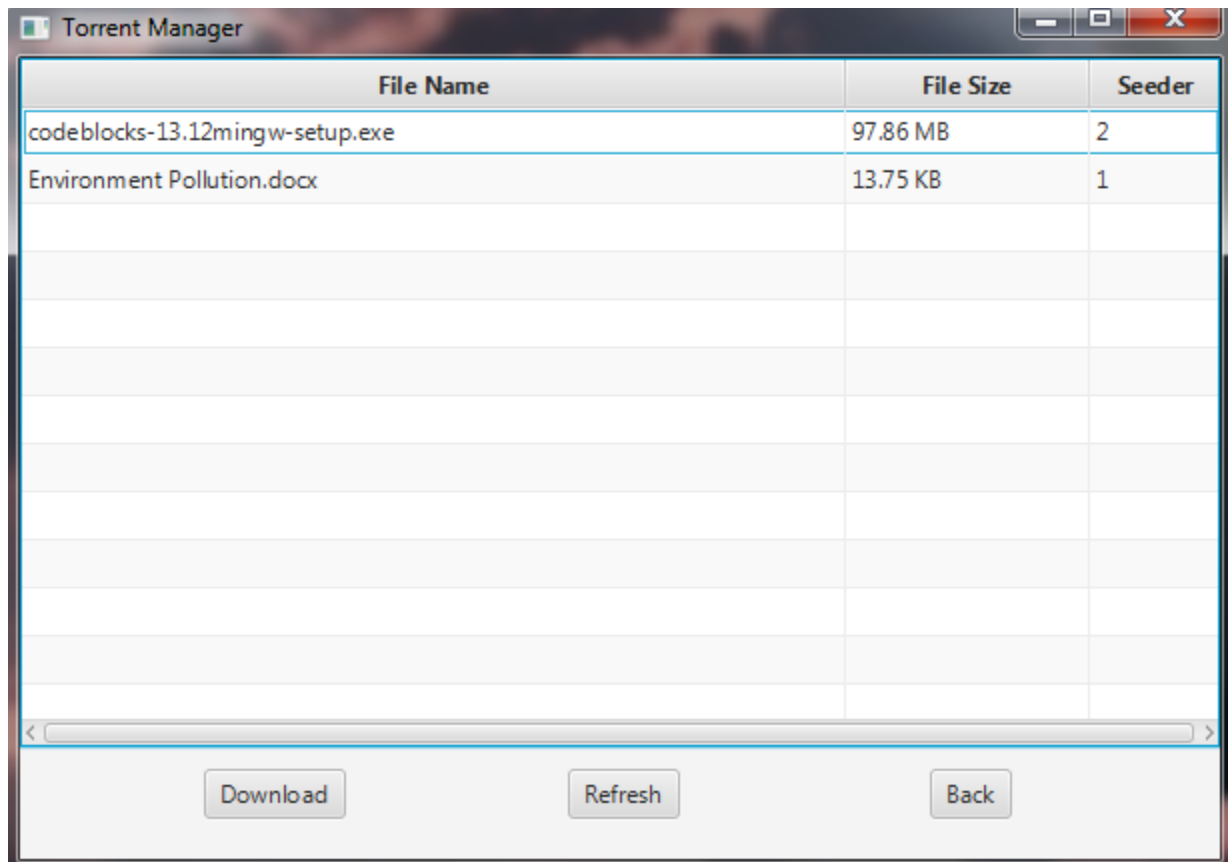


For the user side of this system, at first there are three fields. The CLIENT PORT field expects an integer in 10000-65530 range, where other users will try to connect if this user seeds any file. The SERVER PORT and SERVER IP has to be filled properly according to the server that the user wants to connect. If the user doesn't fill in these fields properly, or such a server system is not found, several warnings and error dialogs are displayed according to the possible mistakes from the user so that the user can try to fill in the fields again and get connected.

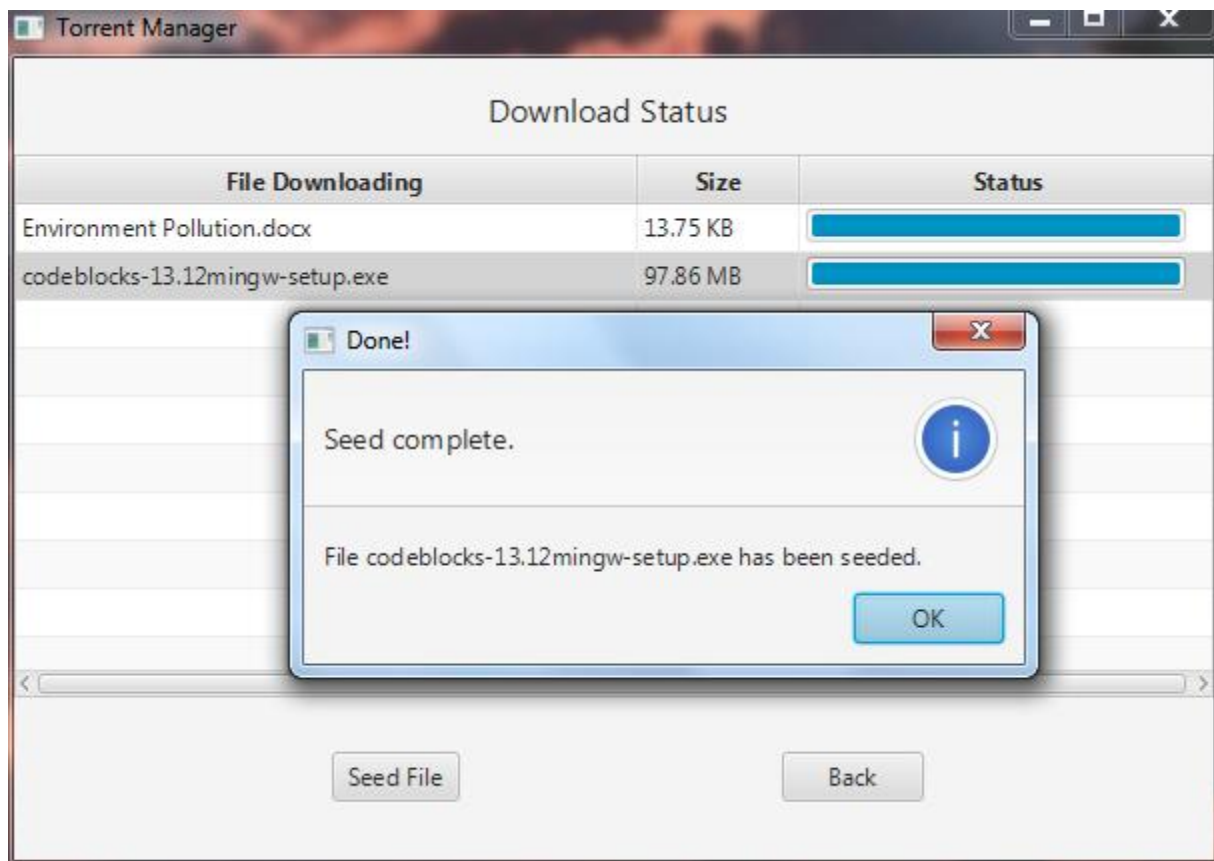




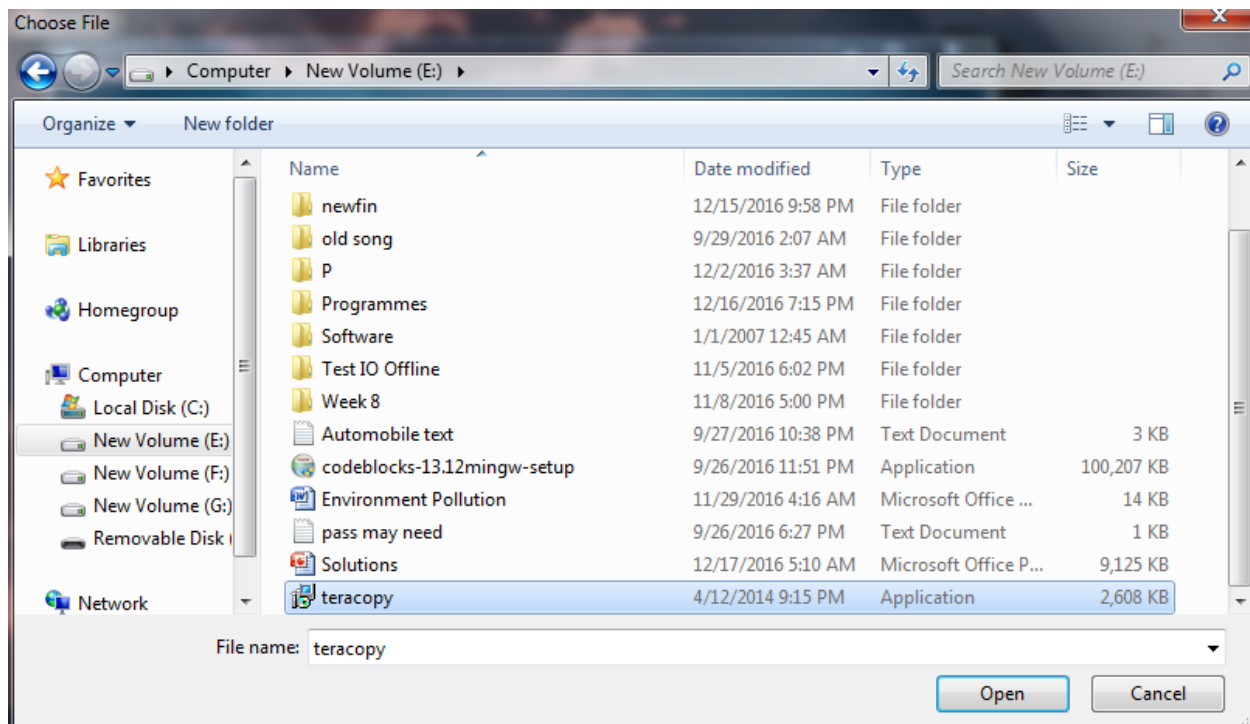
The user can download any file from the file list of the server. So, the file list button loads a page for the user where all the files that have been added to the server are listed. Along with the name of the files, there is file size and number of seeders. A user can select any of these files and press the download button. The user also has to specify the directory for this particular file. Then the download will take place. The back button here takes the user back to the control page.



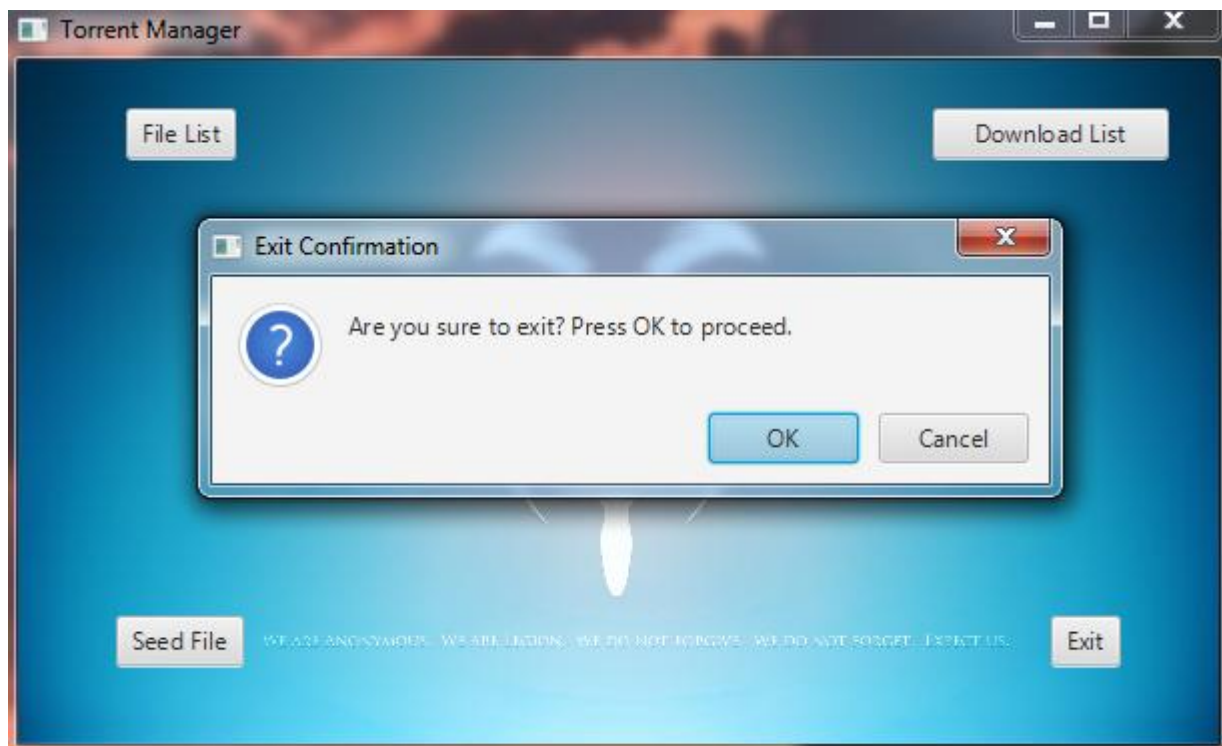
The Download List button loads a page for the user where all the files that have been selected for download in this session, as in after logging in this time. So, if a user downloads some files and exits, after getting connected to the server, the download list will become empty. Once the downloading of a file is complete, user can select the file and seeds it for the other users by pressing the Seed button. The back button takes the user back to the control page.



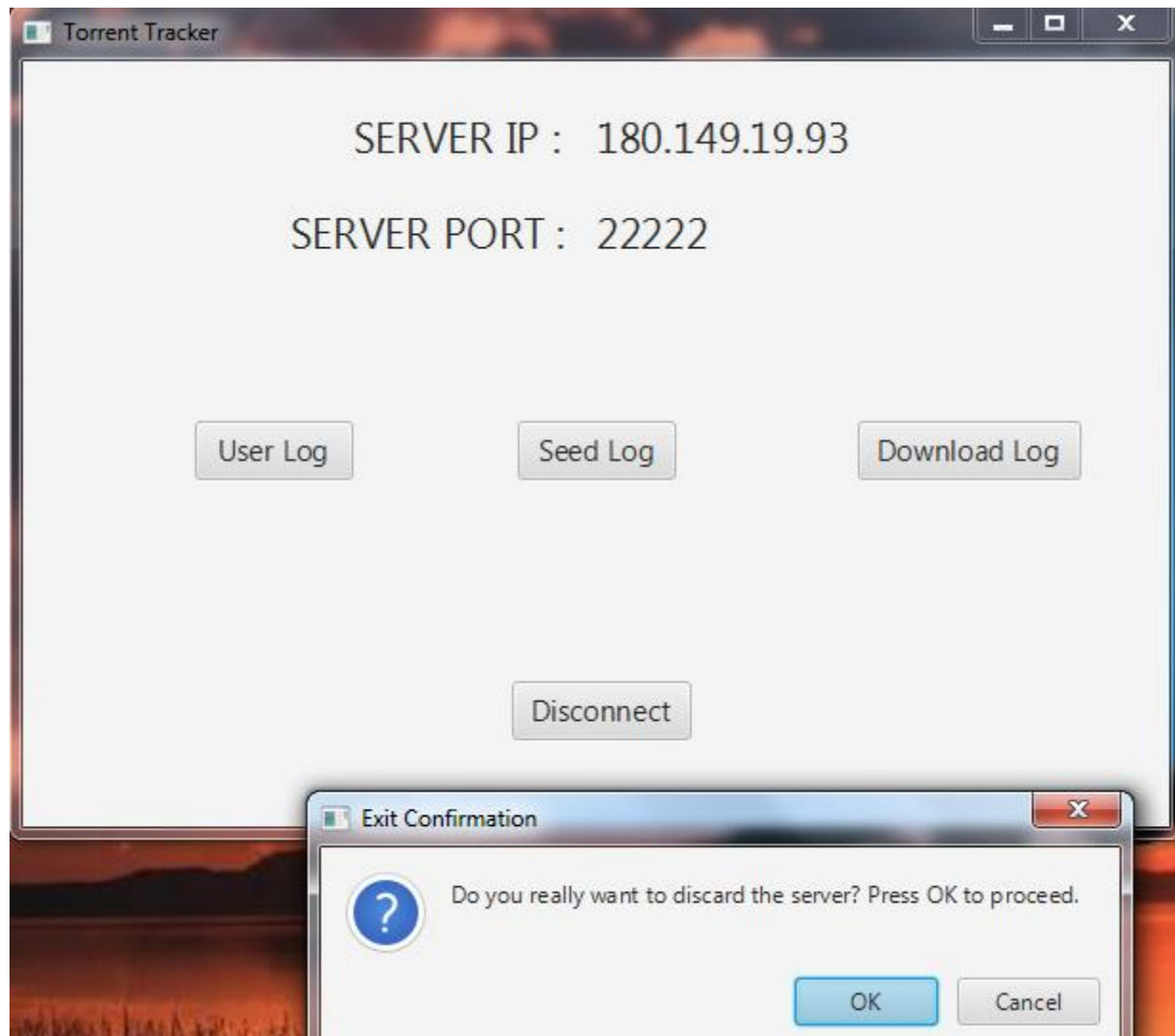
A user can seed files from his/her device also by pressing the Seed button from the control page. It will simply ask for the file that the user wants to seed.



The exit button leads a user towards getting disconnected from the server. A confirmation dialog pops up with an OK and a Cancel button. Pressing OK will exit the system.



The server window is basically for keeping the log of separate actions by users. So, there is no activity for the server. There is a disconnect button if the server needs to be shut down intentionally for some reasons, which is rare in real life scenario. However, the server might get shut down for unknown problems in real Torrent Download system, so handling the appropriate response for the user if this disconnect is pressed ensures proper response for such kind of real life scenario.



Tools I have used:

I have used IntelliJ IDEA 2016.3.1 and JDK 1.8 for my project development. I also used FXML files and JavaFX Scene Builder 2.0 for GUI.

Lacking:

There were several lacking in this project. One of them was the server/tracker cannot check whether a seeder of a file is still connected or not. Thus, the server cannot count him/her as a seeder when s/he is not connected. Thus, server cannot store info into a file, which causes the problem that server always starts from blank state where no file has been seeded yet. Someone has to seed something every time the server is re-launched so that that file can be downloaded by others. This problem could be avoided by more rigorous networking exception handling but it required some more time. As I started working on this project quite late, I could not add this feature which would make this project more relatable to real torrent system.

Source Code Link:

<https://drive.google.com/open?id=0B3kJ6JlcOPEbHVzZ2tLRnBDTIU>

From this folder, by running the 'Main' classes from both the Tracker_GUI and User_GUI packages, anyone can run this system.

The Tracker_GUI.Main launches the tracker system and the User_GUI.Main launches the user system.