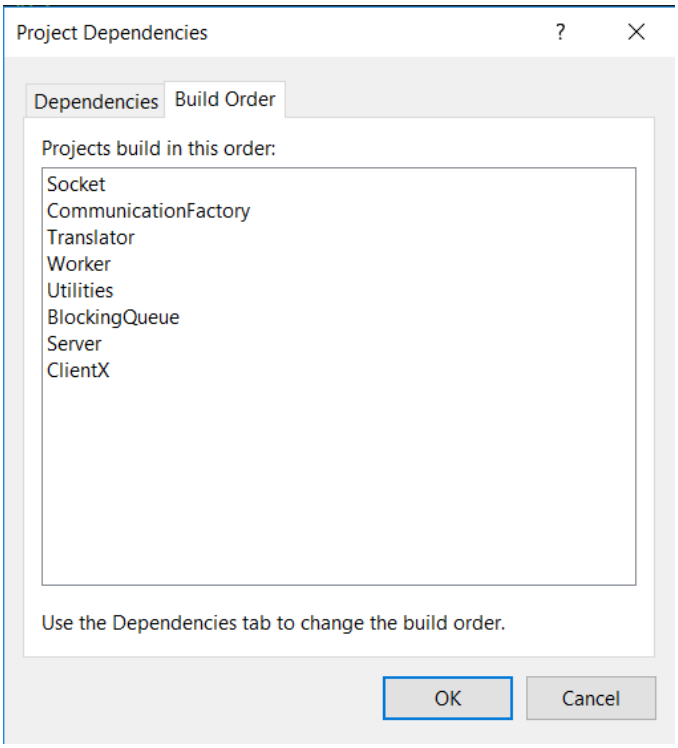
Project #4 OVERVIEW

There are three main executables in this project

1. Server.exe, GUIClient.exe and Worker.exe

After compiling (either Debug or Release), it's best to navigate to the folder and run the applications from there. Note: we have found that on occasion Visual Studio will not rebuild the entire solution. In this case build the following projects:



First, run Server.exe. It'll spawn two Worker.exe processes and show that they registered.

Then, run GUIClient.exe.

Note: If you run GUIClient.exe by itself, it'll start up and display a login screen. You can keep this open until the server is up and running before entering the credentials. If you do enter some credentials before the server comes up, the GUI waits for up to a minute before returning the controls back to the user. Without a spinning wheel to indicate processing, it might appear that the GUI is stuck, but basically it's just spinning and waiting for the server to respond.

An Extra Code Item:
The credentials are stored in a base64-encoded file called authentication.txt that is part of the package and is co-located in the same folder as the server.exe binaries, or rather must be co-located with the server.exe binaries.

If you decode the base64 string, you'd see that there is a default admin user with the given credentials:

```
username: admin
password: this!s$p@rta
```

Special Note: When running the binaries be sure to run as an administrator to avoid any local authentication issues.

- Use these credentials to login for the first time.
- In the login screen, at the bottom, there is a check box that reads 'Open User Management'.
- Checking that box while logging in will allow you to add additional users.

- Note that a user once added, can't be deleted through the GUI. The only way to do it right now is to open the authentication.txt file in the server and delete all the lines except for the first line (which is the default admin user).

FEATURES COMPLETED

1. **Support uploading files** - Use the Upload button below the directory list to upload a file to the current directory.
2. **Support creating remote directories** - Use the 'New Folder' button below the directory list to create a new directory.
3. **Support navigating and displaying remote Repository.**
4. **Support browsing remote logs** - Select a log file (from the TESTRESULTS folder, usually after tests are executed, or you could copy-paste/upload a file) and hit the 'View' button. This button should only be enabled for files with extension .log, .txt & .xml.
5. **Support selecting running tests on DLLs -** To elaborate and gave a little more information on this one. The Worker.exe expects the test driver dll to implement an ITest interface.

We have provided a sample DLL, Test.dll, that tests two other DLLs, Vehicle.dll & Shape.dll. (This is consistent with the other projects). It should be available in the REPOSITORY/Test1 folder.

Test Execution:
When executing a test, you need to select one test driver, and at least one tested code. For this Test.dll to succeed though, it needs both DLL.s. We decided that an edge case test for success and failure scenarios through selecting just one/both.

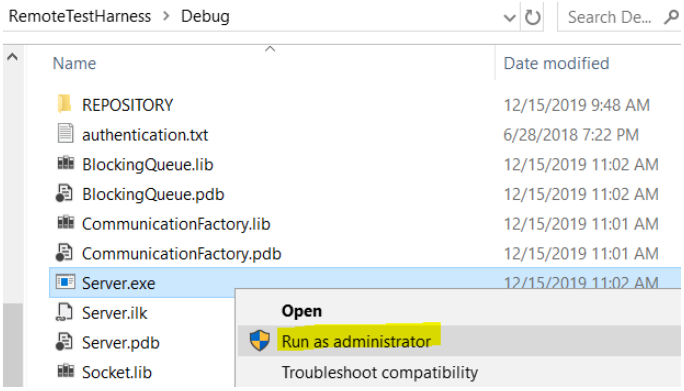
Local package location:

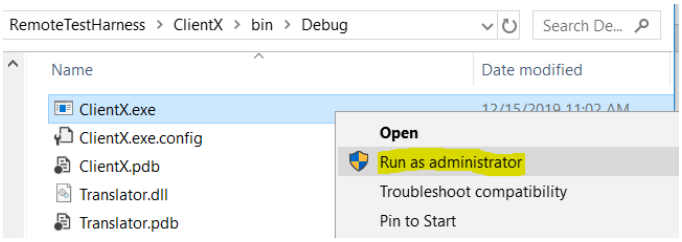
```
Package Build location:
C:\syracuse\Sandbox\CIS_687_ObjectOrientedDesign\Project4\Project4\TestHarness\Debug
C:\syracuse\Sandbox\CIS_687_ObjectOrientedDesign\Project4\Project4\TestHarness\ClientX\bin\Debug
```

Test Execution Steps:

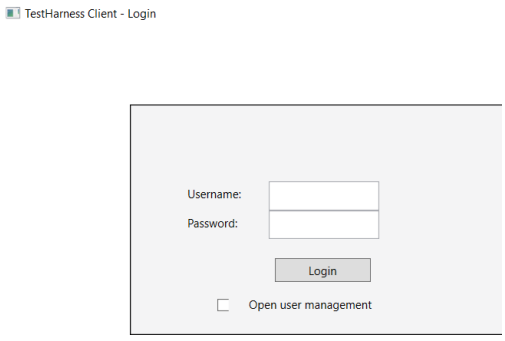
- To execute the test, choose Test.dll in the files list, and then hit the 'Test Driver' button.
- Then proceed to select both Vehicle.dll & Shape.dll, one at a time, and hit the 'Tested Code' button.
- This will enable the 'Execute' button, by virtue of having selected a test driver a tested code.
- Hit execute, and the server sends an acknowledgement, which also states where the test log will be available.
- From this point, the server forwards the request to one of the workers, and have it execute the tests, and send the results back.

1. If the binaries are run individually as stated earlier be sure to run as an administrator

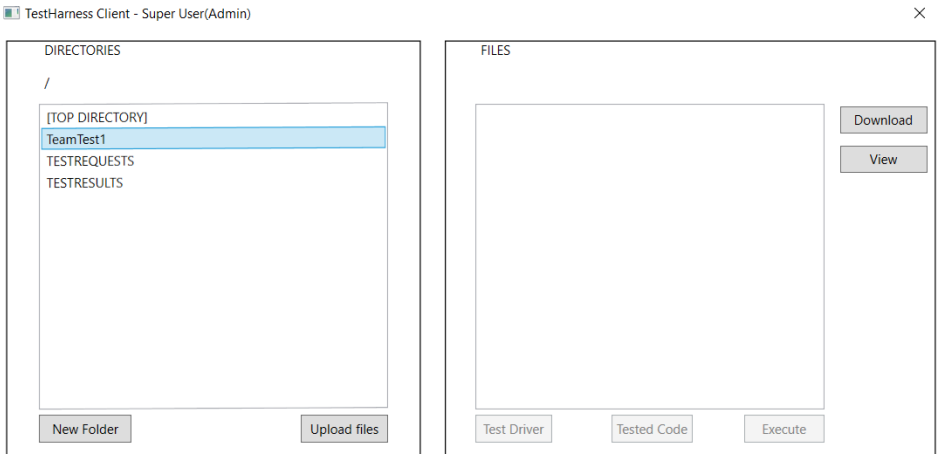




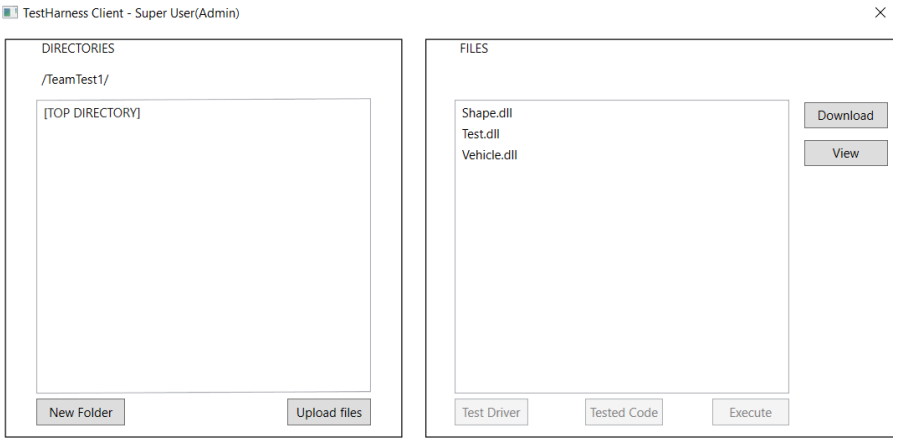
2. Once the client UI is displayed enter the authentication credentials



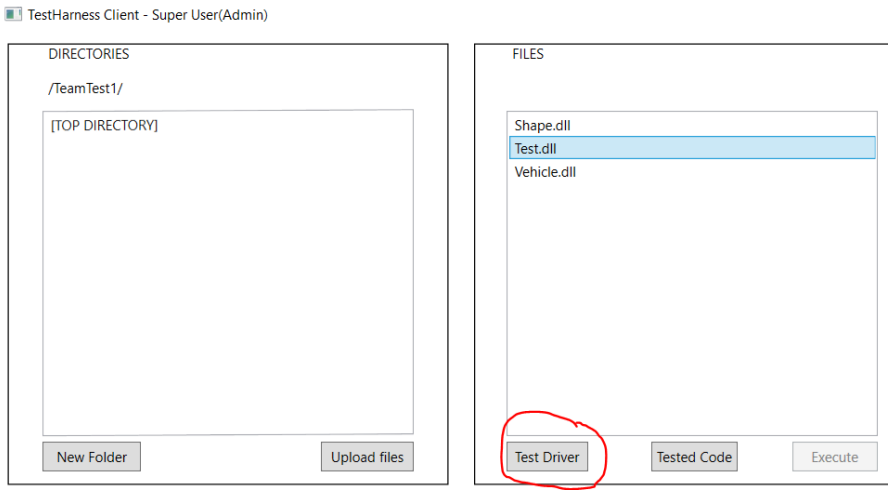
3. Once the repositories have been displayed within the client UI proceed to select the test folder where the test DLL's reside.



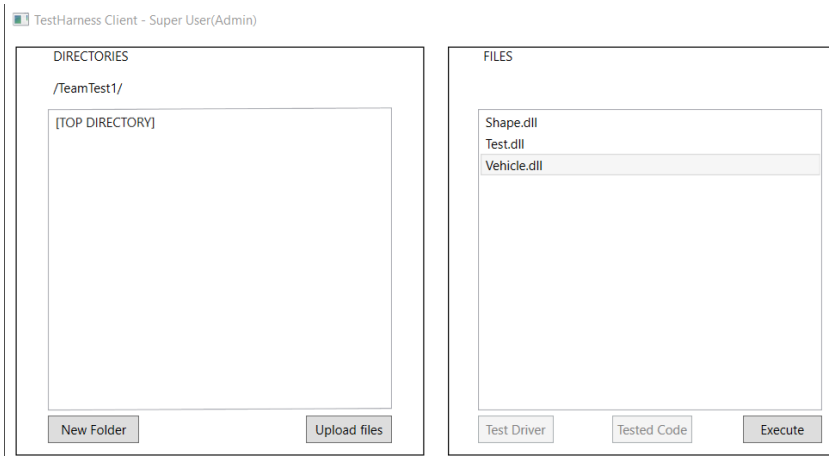
4. Double click on the "Team Test1 folder". The test DLL's will be displayed.



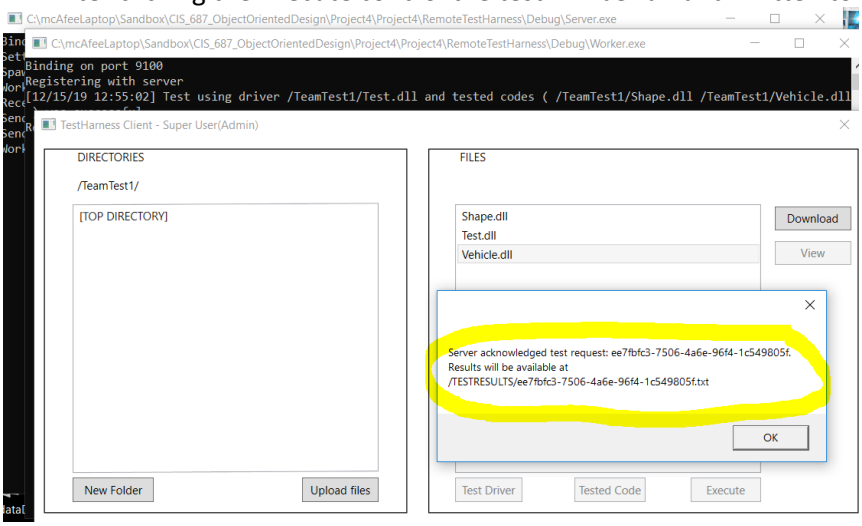
5. Select the test driver DLL



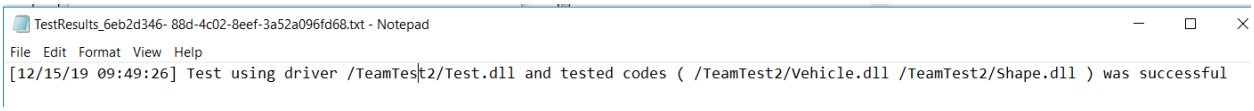
6. This step is key, first highlight Shape.dll the invoke the Tested Code control. Next highlight the Vehicle.dll and invoke the Tested Code control again. For each case the Execute control button will be activated.



7. After clicking the Execute control the test will be run and written to the given file location



8. Sample of test results:



BONUS FEATURES COMPLETED

1. **Support users** entering the system with a **username and a password**, and only a user with **administrator privileges** shall be able to **create new users** - as said before, start with the default admin user, and create users as needed.

2. **Support the downloading of remote files** to a local directory in the client. I don't have support for downloading entire directories, but the original requirement said, 'Download a file **OR** a complete directory', so I went for the first option.