**Ain Shams University**
**Faculty of Engineering**
**Computer and Systems Engineering Department**

**CSE 411: Distributed Computer Systems – 4th Year CSE – 1st Semester 2019/2020**

# PROGRAMMING PROJECT

This is a group programming project with a group of 4 to 6 students. In this project, you are going to apply the programming techniques related to socket and thread programming to build a File Storage server and the corresponding client.

This system provides at least the following features to its client:

1. Create a new user account, **where:**
    a. The user should provide the email address and account password , in order to create the account. The server shouldn't accept duplicate emails.
    b. A unique account ID must be generated for this user by the server, the server should create a **root directory** named with the corresponding unique user ID on the server side.
    c. Each root directory for each client contains a **home directory** to hold this client files and directories. Note that the client top level directory to be viewed is his home directory.
2. Login using the user email and password. **Note:** You must handle invalid email or account password.
3. You should support the change directory command (***cd***) , **where:**
    a. The client should be able to navigate **within** his directories.
    b. The the top level directory that the client could navigate to is his home directory.
4. You should support the print working directory command (***pwd***) , **where:**
    a. The client should be able to check the current directory path.
    b. **Note:** The top level for the current directory path is the client's home directory. You must hide the server main storage details.
5. You should support the list command (***ls***) , **where:** The client should be able to list all files and directories in the current working directory.
6. You should support the make and remove directory commands (***mkdir*** and ***rmdir***) , **where:**
    a. The ***mkdir*** command enables the user to create new directories to current working directory.
    b. The ***rmdir*** command enables the user to remove directories from current working directory.
    c. **Note:** Your system should not accept duplicate name for files or directories within the same parent directory.

7. You should support move,copy,rename and delete file commands (***mv***,***cp***,***rnm*** and ***rm***) , **where:**
    a. The client should be able to manipulate his files moving(***mv)*** or copying(***cp)*** a file to another directory, renaming(***rnm)*** a file within the current directory, or deleting(***rm)*** a file from the current directory.
    b. **Note:** Your system should not accept duplicate name for files or directories within the same parent directory.
8. You should support the upload and download file commands (***upload*** and ***download***) , **where:**
    a. The user could ***upload*** a file from local storage system to the current working directory at the server side.
    b. The user could ***download*** from the current working directory at the server side to the local storage system.
    c. **Note:** Your system should not accept duplicate name for files or directories within the same parent directory.

The project should also fullfil the following:

1. The program should be written in Java.
2. The server should be multitasking that use dedicated thread for each connected client.
3. The server must maintain its state, list of users and their info ,and user's files if it was shutdown or restarted.
4. Exception handling must be used to handle any kind of expected failures.
5. Additionally, It is rquired to submit a detailed report and a demo video using **multiple physical machines for server and clients over a LAN showing all the required use cases**.
6. Expect also a demo session with the TA in order to discuss and examine the work and the functionality of the overall system.

**Extra Credit:**

- Graphical user interface for the client side **(instead of the command line)** application (**10%**).
- Adding a Database to the system **(for user informations and root directories)** (**15%**).
- Applying Security techniques to the system, such as Encrypting/Decrypting sensitive data before storing to the data base, or transmission over the network (**10%**).

**Project Deliverables:**

All Project Deliverables must be sent to the corresponding Google forms.

1. Project proposal (pdf file) containing team members (4-6): full names in arabic and student codes.

   Deadline: Thursday 7/11/2019, 11:59 PM

   Google Form Link: https://forms.gle/oBWGhs7xLwSNkRA27

2. System Design (pdf file)
   - System Architectural Model: showing the computational and communication tasks performed by every computational elements (Client/Server , P2P).
   - The Application Level Protocol adopted in the project. Showing exact messages transmited between client and server organized by use cases.

   Deadline : Thursday 21/11/2019 , 11:59 PM

   Google Form Link: https://forms.gle/w3gRXziqyUPxzTDz6

3. Final Delivery including:
   - Source Code.
   - Executable files (Jar files)
   - A YouTube demonstration video.
   - Project Report:
     i. Final System design (including any updates/changes).
     ii. User Guide with snapshots demonstrating how to successfully perform all the required tasks of the project.
     iii. Any additional documentation you might find useful (including code documentation, descriptions of difficulties encountered, etc.)

   Deadline : Thursday 5/12/2019 , 11:59 PM

   Google Form Link: https://forms.gle/buMagjodeuHDTJaD9

---