Appendix

First Client Interview

Hello, how are you?

I'm alright, how are you?

Good. So, to start the interview, what is the problem you need me to solve with this program?

I need to find some way for my students to have access to their assignments while also maintaining their privacy. We currently have to share files on public school computers and I'd like for the process to be more secure.

I can see why that would be an issue. If I program a file storage system, would that satisfy your needs?

Yes, would that be a private way of file sharing?

It could be. I could make user accounts so that the students would have access to only their own files.

That sounds good, how would I be able to manage those accounts? And how would they log in?

I'm thinking there would be a login prompt at the launch of the program, asking for the student's username and password. To manage the accounts, would an administrator portal work?

Hm… could you elaborate on how the administrator portal would work?

I'm thinking you could log in through the same page as the students, but you'd be taken to a different part of the program where you could create, modify, and delete student accounts.

I see. Would I create the accounts for myself and the students or would the students create their own accounts?

I'll probably make the program with a pre-set admin username and password that I would give you, and you can add in the students' accounts yourself.

How would I know how to create those accounts? Will there be a specific page in the program where I can do that? Would I be provided with a tutorial or instructions?

Instructions are a good idea. When I write the program I'll make sure that everything is comprehensive. As of right now, I'm planning on having prompts show up for different things the users and admin can do, so they are easily selectable. Like, for example, "would you like to do x, y, or z," and you can respond with one of those three options. The specific page in the program where you'd create the accounts would be in the administrator portal we were discussing earlier.

So what other features can you think of that you'd want in the program?

I would need a way to post assignments that my students have access to.

Yeah, of course. I can integrate that into the administrator portal. I was also thinking about integrating some form of encryption into the program. What do you think of that?

That sounds good. What exactly will that prevent them from accessing?

Every student's files are inaccessible to others because of the login. However, they would still be able to access private assignments by going to the location on the computer where the program stores them and opening the files. Encrypting the assignments would solve that issue, as the files wouldn't be comprehensible. The students won't be able to see unencrypted files unless they log into their own accounts.

Great, I hadn't even considered that students would be still able to see assignments if they weren't logged into the program.

Any other additional features you would want to program to have?

Maybe a way to make sure the students are automatically logged out their files are secure in case they forget to log out before leaving the computer. Other than that, it sounds great!

All right, I'll make sure that's in the final product. If that's all, have a great rest of your day.

Thanks, you too!

Final Client Interview

Hello again, it's been a while.

Yes, it has. I looked over the product you gave me.

What are your thoughts on it?

Overall I am very content with the work you did, it checks all of the boxes for what I remember us talking about in our last interaction.

Thank you, I tried to use them as a checklist so as to make sure you got what you wanted.

Yes, it shows. Although, even though you did everything I asked for, I kind of wish there was more. I didn't completely intend for what we talked about to be the extent of the program and thought you'd add more features on your own.

I apologize for the misunderstanding, I'll keep that in mind for the future. Speaking of other features I could have included, what are some things you'd recommend changing or adding to the program for later development?

Hm… an interface would be nice, similar to other modern applications. The console format is adequate for a program like this since it's mainly a file viewer, but it might be easier for the students to use buttons to interact with the application.

Thanks for the suggestion. Anything else?

I think it would have been nice to have certain students barred from seeing certain assignments. Even though my class has a standard set of assignments so there is no issue with sharing, other teachers could use the same program on the public computers to distribute assignments to their students, so controlling who sees certain assignments might be useful. Also, speaking of other teachers using the program, it would be nice to have teacher accounts separate from the administrator and student accounts, that could create and modify files, but not access or modify the student accounts.

Those are some interesting ideas. Thank you for your input.

Of course. I think that's all I can think of for now. Thank you!

Have a good day!

Program

Main.java:

```
package com.company;

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {

        AdminCommands.setUpFilenames();
        System.out.println("Welcome to the encrypted file storage system!");

        while (true) {

            System.out.println("\nPlease input the following information to log
in, or type \"end\" to end the program.");

            System.out.println("Username:");
            Scanner usernameScanner = new Scanner(System.in);
            String usernameString = usernameScanner.nextLine();

            if (usernameString.equals("end")) System.exit(0);

            System.out.println("Password:");
            Scanner passwordScanner = new Scanner(System.in);
            String passwordString = passwordScanner.nextLine();

            if (passwordString.equals("end")) System.exit(0);

            AdminCommands.decodeAll();

            if (usernameString.equals("admin") &&
passwordString.equals("adminpass")) {

                AdminCommands.encodeAll();

                System.out.print("\nAdmin Login Successful.");

                admin:
                {

                    while (true) {

                        while (true) {
```

```
                    System.out.println("\nA list of actions you can run
is printed below.");
                    System.out.println("\n- Edit User Accounts\n-
View/Edit Assignment Files\n");
                    System.out.println("Please type the name of the
action you would like to run, or type \"exit\" if you would like to log
out.");

                    Scanner adminScanner = new Scanner(System.in);
                    String adminString;

                    while (true) {

                        adminString = adminScanner.nextLine();

                        if (adminString.equals("exit")) {
                            break admin;
                        }

                        if (adminString.equals("Edit User Accounts") ||
adminString.equals("View/Edit Assignment Files")) {
                            break;
                        }

                        System.out.println("This is not a valid action.
Once again the list of actions you can run is printed below.");
                        System.out.println("\n- Edit User Accounts\n-
View/Edit Assignment Files\n");
                        System.out.println("Please type the name of the
action you would like to run, or type \"exit\" if you would like to log
out.");

                    }

                    if (adminString.equals("Edit User Accounts")) {

                        System.out.println("\nWould you like to edit
the list of usernames or passwords?");

                        while (true) {

                            adminString = adminScanner.nextLine();

                            if (adminString.equals("usernames") ||
adminString.equals("passwords")) {
                                break;
                            }
```

```
                                    System.out.println("This is not a valid
option. Would you like to edit the list of usernames or passwords?");
                                }

                                if (adminString.equals("usernames")) {

                                    while (true) {

                                        System.out.println("\nThe usernames
currently saved on this computer are:\n");
                                        AdminCommands.printUsernames();
                                        System.out.println("\nWould you like to
edit this list? (Answer with 'y' or 'n')");
                                        adminString = adminScanner.nextLine();

                                        if (adminString.equals("y")) {

                                            System.out.println("\nPlease retype
the list of usernames with the changes you'd like, with a space between each
username.");
                                            adminString =
adminScanner.nextLine();


AdminCommands.changeUsernames(adminString);

                                            System.out.println("\nUsernames
successfully changed.");

                                        } else break;

                                    }

                                } else {

                                    while (true) {

                                        System.out.println("\nThe passwords
currently saved on this computer are:\n");
                                        AdminCommands.printPasswords();
                                        System.out.println("\nWould you like to
edit this list? (Answer with 'y' or 'n')");
                                        adminString = adminScanner.nextLine();

                                        if (adminString.equals("y")) {

                                            System.out.println("\nPlease retype
the list of passwords with the changes you'd like, with a space between each
password.");
```

```
                                                adminString =
adminScanner.nextLine();

AdminCommands.changePasswords(adminString);

                                                System.out.println("\nPasswords
successfully changed.");

                                            } else break;

                                        }

                                    }

                                } else {

                                    while (true) {

                                        System.out.println("\nWould you like to
view, edit, create, or delete a file? If not, type exit.");
                                        adminString = adminScanner.nextLine();

                                        if (adminString.equals("view")) {

                                            System.out.println("\nThe files
currently stored on this machine are:\n");
                                            AdminCommands.printFiles();

                                            System.out.println("\nWhich file would
you like to view?");
                                            adminString = adminScanner.nextLine();

                                            System.out.println("\nThe content
stored in this file is:\n");

AdminCommands.printContent(adminString);

                                            break;

                                        } else if (adminString.equals("edit")) {

                                            System.out.println("\nThe files
currently stored on this machine are:\n");
                                            AdminCommands.printFiles();
                                            System.out.println("\nWhich file would
you like to edit?");
                                            adminString = adminScanner.nextLine();
```

```
                                        System.out.println("\nThe text
currently stored in the selected file is:\n");

AdminCommands.printContent(adminString);

                                        System.out.println("\nPlease enter the
new text to be saved to the file:");
                                        String adminString2 =
adminScanner.nextLine();
                                        AdminCommands.writeFile(adminString,
adminString2);

                                        System.out.println("\nThe new text
saved in the file is:\n");

AdminCommands.printContent(adminString);

                                        break;

                                    } else if (adminString.equals("create")) {

                                        System.out.println("\nWhat is the name
of the file you want to create? (append the name with \".txt\")");
                                        adminString = adminScanner.nextLine();

                                        System.out.println("\nPlease enter the
content of your new file:");
                                        String adminString2 =
adminScanner.nextLine();

                                        AdminCommands.createFile(adminString,
adminString2);

                                        System.out.println("\nFile created.");

                                        break;

                                    } else if (adminString.equals("delete")) {

                                        System.out.println("\nA list of the
files on the system is printed below:\n");
                                        AdminCommands.printFiles();

                                        System.out.println("\nWhich of these
files would you like to delete?");
                                        adminString = adminScanner.nextLine();

                                        AdminCommands.deleteFile(adminString);
```

```
                                        System.out.println("File deleted.");

                                        break;

                                    } else if (adminString.equals("exit")) {

                                        break;

                                    } else System.out.println("This is not a
valid input.");

                                }

                            }

                        }

                    }

                } else {

                    boolean login = Authenticator.checkLogin(usernameString,
passwordString);

                    AdminCommands.encodeAll();

                    if (login) {
                        break;
                    }

                    System.out.println("\nIncorrect login information. Please try
again.");
                }

            }

            System.out.println("\nLogin successful.");

            while (true) {

                System.out.println("A list of files that you have access to is
printed below.");
                System.out.println("\n- assignment1.txt\n- assignment2.txt\n");
                System.out.println("Please type the name of the file that you wish
to view.");

                Scanner fileScanner = new Scanner(System.in);
```

```
            String fileString;
            while (true) {

                System.out.println("File:");
                fileString = fileScanner.next();

                if (fileString.equals("assignment1.txt") ||
fileString.equals("assignment2.txt")) {
                    break;
                }

                System.out.println("\nThis is not a valid filename. Once again
a list of files that you have access to is printed below.");
                System.out.println("\n- assignment1.txt\n- assignment2.txt\n");
                System.out.println("Please type the full name of the file that
you wish to view.");

            }

            System.out.println("\nYou have selected the file " + fileString);
            System.out.println("The content of this file is:");

            File file = new File("Resources/" + fileString);
            Path filePath = Path.of("Resources/" + fileString);
            String content = Files.readString(filePath);

            System.out.println(content);

            System.out.println("\nThis information is currently encrypted.
Would you like to decrypt it? (Answer with 'y' or 'n')");

            Scanner yornScanner = new Scanner(System.in);
            String yornString;

            while (true) {

                yornString = yornScanner.next();

                if (yornString.equals("y")) {
                    break;
                }

                System.out.println("\nWould you like to view another file?
(Answer with 'y' or 'n')");
                yornString = yornScanner.next();

                if (yornString.equals("n")) {
                    continue;
                }
```

```
                System.out.println("\nThe developer thanks you for using this
program. All of the decrypted files are already re-encrypted for security, and
you will be logged out upon exiting the program.");
                System.exit(0);
            }

            Encrypt.decode(("Resources/" + fileString), ("Resources/" +
fileString));
            System.out.println("\nThe file has been decrypted. Here is the
content stored in it:");
            content = Files.readString(filePath);
            System.out.println(content);

            Encrypt.encode(("Resources/" + fileString), ("Resources/" +
fileString));

            System.out.println("\nWould you like to view another file? (Answer
with 'y' or 'n')");

            while (true) {

                yornString = yornScanner.next();

                if (yornString.equals("y")) {

                    break;

                }

                System.out.println("\nThe developer thanks you for using this
program. All of the decrypted files are already re-encrypted for security, and
you will be logged out upon exiting the program.");
                System.exit(0);

            }

        }

    }

}
```

AdminCommands.java:

```
package com.company;

import java.io.File;
import java.io.IOException;
```

```java
import java.io.PrintWriter;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.List;

public class AdminCommands {

    public static List<String> filenames = new ArrayList<String>();

    static void setUpFilenames() throws IOException {

        filenames.add("assignment1.txt");
        filenames.add("assignment2.txt");

    }

    static void printFiles() throws IOException {

        String fileString = filenames.toString();
        fileString = fileString.replaceAll("\\[", "");
        fileString = fileString.replaceAll("]", "");
        fileString = fileString.replaceAll(", ", "\n");
        System.out.println(fileString);

    }

    static void addFiles(String fileName) throws IOException {

        filenames.add(fileName);

    }

    static void printContent(String fileName) throws IOException {

        Path filePath = Path.of("Resources/" + fileName);
        Encrypt.decode("Resources/" + fileName, "Resources/" + fileName);
        String content = Files.readString(filePath);
        Encrypt.encode("Resources/" + fileName, "Resources/" + fileName);
        System.out.println(content);

    }

    static void writeFile(String fileName, String content) throws IOException {

        try (PrintWriter out = new PrintWriter("Resources/" + fileName,
StandardCharsets.UTF_8)) {
            out.write(content);
```

```
        }

        Encrypt.encode("Resources/" + fileName, "Resources/" + fileName);

    }

    static void createFile(String fileName, String content) throws IOException
{

        File file = new File("Resources/" + fileName);

        try (PrintWriter out = new PrintWriter("Resources/" + fileName,
StandardCharsets.UTF_8)) {
            out.write(content);
        }

        Encrypt.encode("Resources/" + fileName, "Resources/" + fileName);

        filenames.add(fileName);

    }

    static void deleteFile(String fileName) throws IOException {

        Path filePath = Path.of("Resources/" + fileName);
        Files.delete(filePath);

    }

    static void encodeAll() throws IOException {

        Encrypt.encode("Resources/usernames.txt", "Resources/usernames.txt");
        Encrypt.encode("Resources/passwords.txt", "Resources/passwords.txt");

    }

    static void decodeAll() throws IOException {

        Encrypt.decode("Resources/usernames.txt", "Resources/usernames.txt");
        Encrypt.decode("Resources/passwords.txt", "Resources/passwords.txt");

    }

    static void printUsernames() throws IOException {

        Path filePath = Path.of("Resources/usernames.txt");
        Encrypt.decode("Resources/usernames.txt", "Resources/usernames.txt");
        String users = Files.readString(filePath);
        Encrypt.encode("Resources/usernames.txt", "Resources/usernames.txt");
```

```
        System.out.println(users);

    }

    static void printPasswords() throws IOException {

        Path filePath = Path.of("Resources/passwords.txt");
        Encrypt.decode("Resources/passwords.txt", "Resources/passwords.txt");
        String passes = Files.readString(filePath);
        Encrypt.encode("Resources/passwords.txt", "Resources/passwords.txt");
        System.out.println(passes);

    }

    static void changeUsernames(String changeUsernamesInput) throws IOException
{

        File file = new File("Resources/usernames.txt");

        String changedUsernames = changeUsernamesInput.replaceAll(" ", "\n");

        try (PrintWriter out = new PrintWriter(file, StandardCharsets.UTF_8)) {
            out.write(changedUsernames);
        }

        Encrypt.encode("Resources/usernames.txt", "Resources/usernames.txt");

    }

    static void changePasswords(String changePasswordsInput) throws IOException
{

        File file = new File("Resources/passwords.txt");

        String changedPasswords = changePasswordsInput.replaceAll(" ", "\n");

        try (PrintWriter out = new PrintWriter(file, StandardCharsets.UTF_8)) {
            out.write(changedPasswords);
        }

        Encrypt.encode("Resources/passwords.txt", "Resources/passwords.txt");

    }

}
```

Encrypt.java:

```
package com.company;
```

```java
import java.io.IOException;
import java.io.PrintWriter;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class Encrypt {

    static void encode(String startPath, String endPath) throws IOException {

        Path filePath = Path.of(startPath);
        String content = Files.readString(filePath);
        String[] splitContent = content.split("");
        int forLength = (int) Math.ceil(Math.sqrt(content.length()));
        String[][] originalArray = new String[forLength][forLength];
        String[][] scrambledArray = new String[forLength][forLength];

        int position = 0;
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                if (position >= content.length()) {

                    originalArray[i][j] = " ";

                } else {

                    originalArray[i][j] = splitContent[position];
                    position++;

                }

            }

        }

        position = 0;
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                scrambledArray[i][j] = originalArray[j][i];
                position++;

            }

        }
```

```
        String output = "";
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                output = output + scrambledArray[i][j];

            }

        }

        try (PrintWriter out = new PrintWriter(endPath,
StandardCharsets.UTF_8)) {

            out.write(output);

        }

    }

    static void decode(String startPath, String endPath) throws IOException {
        .
        Path filePath = Path.of(startPath);
        String content = Files.readString(filePath);
        String[] splitContent = content.split("");
        int forLength = (int) Math.ceil(Math.sqrt(content.length()));
        String[][] scrambledArray = new String[forLength][forLength];
        String[][] originalArray = new String[forLength][forLength];

        int position = 0;
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                if (position >= content.length()) {
                    scrambledArray[i][j] = " ";

                } else {

                    scrambledArray[i][j] = splitContent[position];
                    position++;

                }

            }

        }
```

```
        position = 0;
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                originalArray[i][j] = scrambledArray[j][i];
                position++;

            }

        }

        String output = "";
        for (int i = 0; i < forLength; i++) {

            for (int j = 0; j < forLength; j++) {

                output = output + originalArray[i][j];

            }

        }


        try (PrintWriter out = new PrintWriter(endPath,
StandardCharsets.UTF_8)) {

            out.write(output);

        }

    }

}
```

Authenticator.java:

```
package com.company;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;

public class Authenticator {

    static boolean checkUsername(String usernameInput) throws IOException {

        Path filePath = Path.of("Resources/usernames.txt");
        String users = Files.readString(filePath);
```

```
        String[] usersSplit = users.split("\n");

        for (int i = 0; i < usersSplit.length; i++) {
            if (usersSplit[i].equals(usernameInput)) {

                return true;

            }

        }
        return false;

    }

    static boolean checkPassword(String passwordInput) throws IOException {

        Path filePath = Path.of("Resources/passwords.txt");
        String passes = Files.readString(filePath);
        String[] passesSplit = passes.split("\n");

        for (int i = 0; i < passesSplit.length; i++) {
            if (passesSplit[i].equals(passwordInput)) {

                return true;

            }

        }
        return false;

    }

    static boolean checkLogin(String usernameInput, String passwordInput)
throws IOException {

        return checkUsername(usernameInput) && checkPassword(passwordInput);

    }

}
```

assignment1.txt

```
Tx   ?=     hama
 2
++ imas3
?z3   sptsx7  x62  lhi x
-^4  ieeg+y2 2 = s mn  13 =   oam2+xy+ ? aftey ^  0  n in 42=1    act=y  6
 ens.  +?y2
```

assignment2.txt

```
Emt o!nyeDr j r d ociri yroen  incg
```

usernames.txt

```
sesesennararaaretmnmnmnw2papapas1lm1m1mt3eeeeeeu u1u2u3d s
s
s
e
```

passwords.txt

```
sssssnoasasasermwmwmwwdpopopop3lr1r1ra1edededs2p1p2p3s a
a
a
w
```