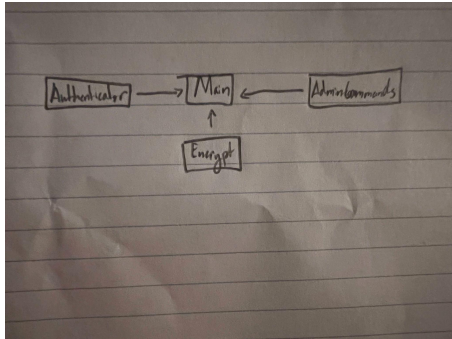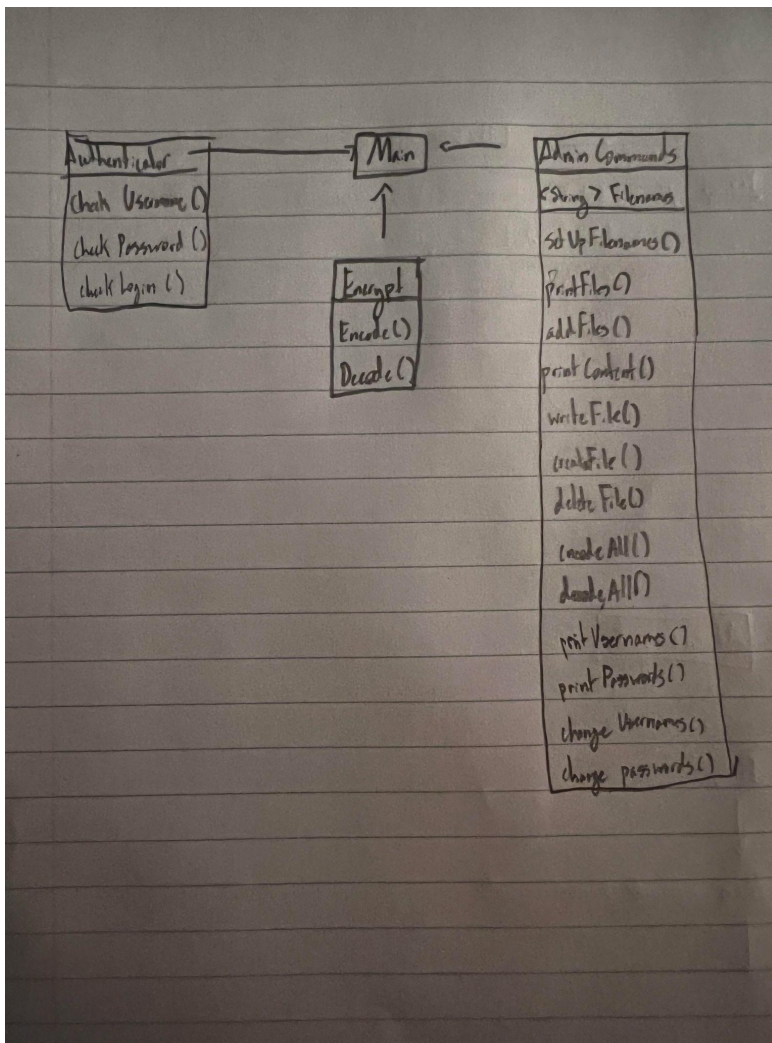# Criterion B - Design

**UML Diagrams:**
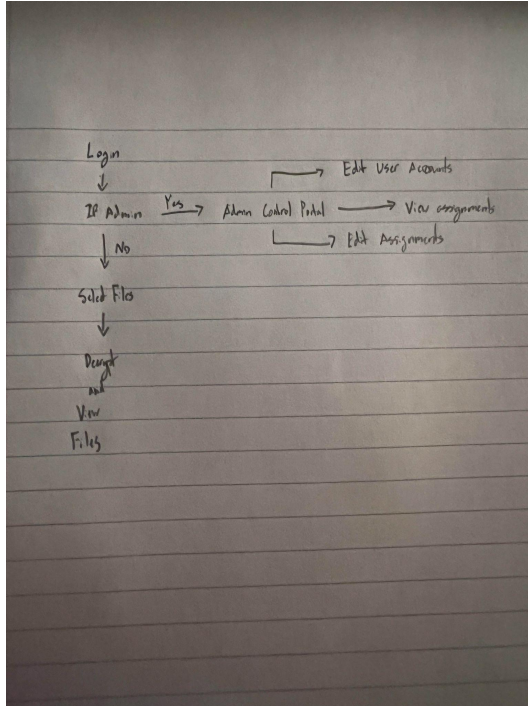
*Stage 1: Basic Structure*



*Stage 2: Full UML Diagram*

**System Flowchart:**



**Pseudocode Algorithms:**

*encrypt():*

content = contents of file at starting path

forLength = round-up of square root of length of content

originalArray = 2D array of size forLength x forLength

scrambledArray = 2D array of size forLength x forLength

position = 0

for i = 0 to forLength-1 do

   for j = 0 to forLength-1 do

      if position >= length of content then

         originalArray[i][j] = " "

      else

```
            originalArray[i][j] = character at position in content

            position = position + 1

        end if

    end for

end for


for i = 0 to forLength-1 do

    for j = 0 to forLength-1 do

        scrambledArray[i][j] = originalArray[j][i]

    end for

end for


output = ""

for i = 0 to forLength-1 do

    for j = 0 to forLength-1 do

        output = output + scrambledArray[i][j]

    end for

end for


write output to file at ending path
```

*decrypt():*

```
content = contents of file at starting path

forLength = round-up of square root of length of content

originalArray = 2D array of size forLength x forLength

scrambledArray = 2D array of size forLength x forLength
```

```
position = 0
for i = 0 to forLength-1 do
    for j = 0 to forLength-1 do
        if position >= length of content then
            scrambledArray[i][j] = " "
        else
            scrambledArray[i][j] = character at position in content
            position = position + 1
        end if
    end for
end for


for i = 0 to forLength-1 do
    for j = 0 to forLength-1 do
        originalArray[i][j] = scrambledArray[j][i]
    end for
end for


output = ""
for i = 0 to forLength-1 do
    for j = 0 to forLength-1 do
        output = output + originalArray[i][j]
    end for
end for


write output to file at ending path
```

**Test Plan:**

| Success Criteria | Testing Procedure | Expected Results |
|---|---|---|
| The program uses a user-friendly GUI | Interviewing the client | Client agrees that the GUI is user-friendly |
| The program allows the persistent storage of text documents (tests) across instances | Opening and closing the program to see if the text documents save across instances | Text documents being saved and accessible across program instances |
| The program employs a sorting system where tests are sorted using filters to determine who can access them | Using student accounts to check if filters correctly give/remove access to different files | Filters show the files to the students that should see them, and hide the ones they shouldn't |
| The program has an account creation portal for students | Using the account creation portal to create an account | Account getting created successfully |
| The accounts created in the program must use an email, that is verified by the program, so that people cannot create more than one account and so that the identity of the account creator is verified. | Run the program using fake emails and valid emails to see if the verification system works | The program declining fake emails and allowing accounts to be created with the real ones |
| The program has a login page for users so that authorized users can access the tests. | Testing logging in with an account | Login page allows authorized users to gain access to the program |
| The program has an administrator access portal that allows the controlling of students' access to different tests. | Logging in with an admin account | Logging in with the admin account opening up the admin page |
| The program has a section for answer keys for the tests so | Checking if the program has answer keys | The program having answer keys |

| | | |
|---|---|---|
| that when the students have already taken the tests they can check their work as review | | |
| The program has a way of encrypting text documents so that hackers cannot understand the files even if they find them in the system | Encrypting the documents and finding them in the computer to see if they are encrypted | Documents being encrypted |
| The program will also be able to decrypt the text documents so that when the students log in they can understand the files | Decrypting the documents and finding them in the computer to see if they are decrypted | Documents being decrypted |