

## Criterion B

### Test Plan:

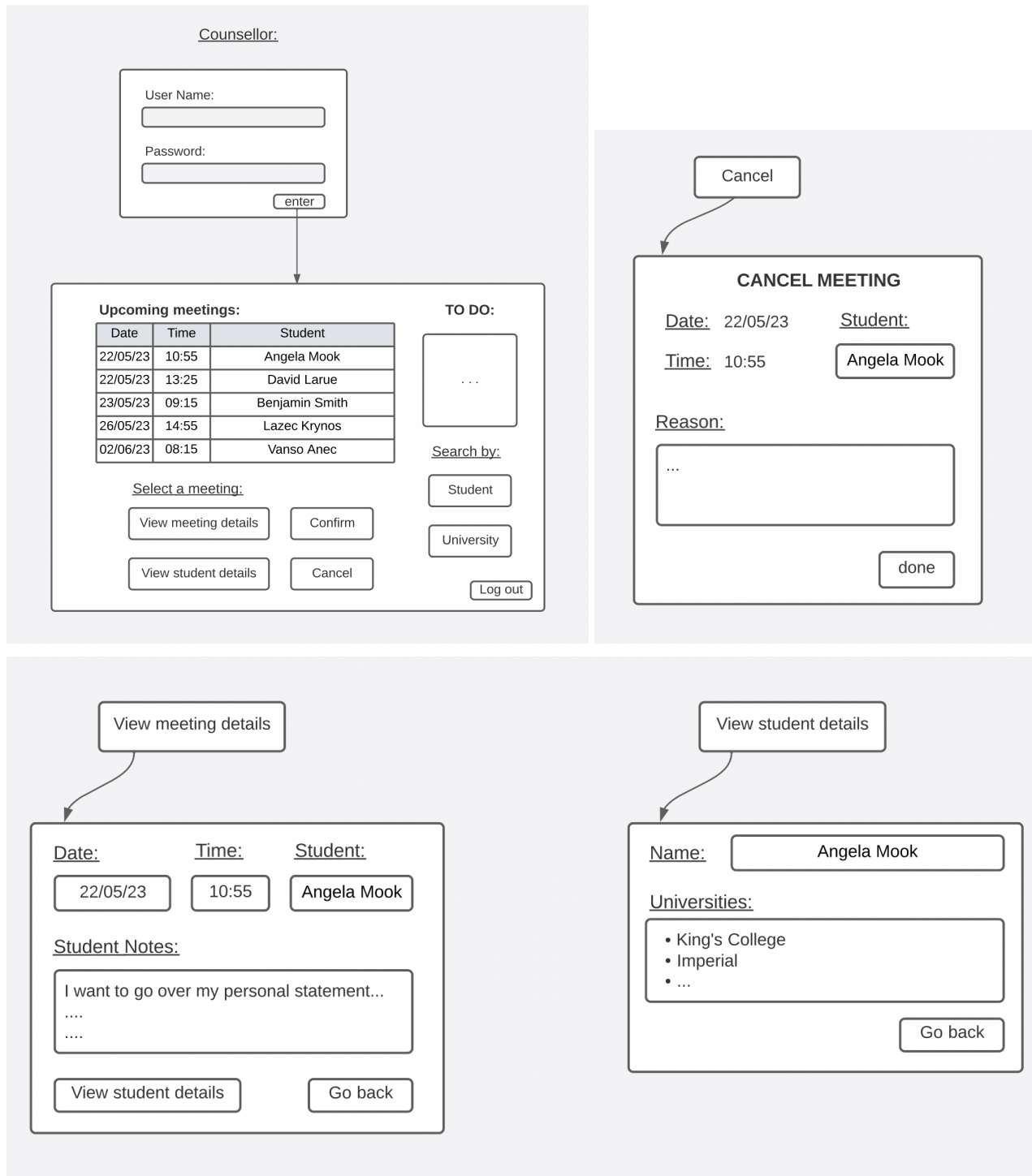
Criteria	Test	Example Data
Users can login as either a counsellor or a student	<p>The user logs into their accounts, displaying the corresponding MainPage. The username and passwords of each account are stored in a text file.</p> <p>When a user makes a new account, it is added to the text file.</p>	<p><b>Student</b> Username: Alex D. Password: 123</p> <p><b>Counsellor</b> Username: Ms. Perky Password: Padua High</p> <p><b>Files:</b> Student class - StuList8.txt Meeting class - AllMeetList8.txt</p> <p>All data is saved to the files when program closes</p>
Students can schedule meetings and input details in a note section	<p>Students are able to schedule meetings at a desired time and date.</p> <p>The student will be able to edit a note section below the details of the meeting. These notes will be displayed to the counsellor through “View meeting details”</p>	<p><b>Alex D. meetings</b> Date: 12.11.2023 Time: 10:30 Notes: UCAS (PS)</p> <p>Date: 14.11.2023 Time: 12:25 Notes: UCAS</p> <p>Details are shown when the counsellor selects a meeting and clicks “View meeting details”</p>
Students can select which universities they want to go to	When checking their account settings, students are allowed to select the universities they plan to apply to.	<p><b>Alex D. universities</b> University of Oxford University College London King’s College London</p> <p>→ Saved when program closes</p>
Counsellors and students can view all meetings in an “upcoming” list	Both of the user interfaces of the student or counsellor will contain a list of their upcoming meetings	<p><b>Student</b> Previously made meetings appear when clicking “View all meetings”.</p> <p>“Upcoming meeting” will contain the earliest meeting in the MainPageStudent.</p> <p><b>Counsellor</b> Previously made meetings appear under “Upcoming meetings” in the MainPageCounsellor</p>

<p>Counsellors can cancel meetings, with the student receiving a warning about the update</p>	<p>When clicking on a meeting, the counsellor has the option to cancel it, with an optional explanation.</p> <p>When the student logs into their account, a pop-up window will appear, informing them of this cancelled meeting.</p>	<p>Counsellor cancels this meeting:  Date: 12.11.2023  Time: 10:30  Notes: UCAS (PS)  Reason: “no longer available”</p> <p>When the student logs in, a window appears with the information of the meeting cancelled, and the counsellor’s reason.</p> <p>The cancelled meeting no longer appears in the upcoming list in the MainPageStudent</p>
<p>Counsellors can search scheduled meetings based on the student and university</p>	<p>Counsellors can successfully search meetings planned by a particular student.</p> <p>Counsellors can search for students wanting to apply to a certain university.</p>	<p>When the counsellor searches for meetings with Alex D. the following appears (before one of the meetings was cancelled):  12.11.2023 - 10:30  14.11.2023 - 12:25</p> <p>When searching for any of the universities we inputted for Alex D., this username will be the output</p>

## GUI Design:

→ The initial plan for how the user interface is going to look like

### Counsellor interface



Cancel

**CANCEL MEETING**

Date: 22/05/23      Student:

Time: 10:55      Angela Mook

Reason:

...

done

Search by:

Student

Angela Mook ▼ Go

Date	Time	Student
22/05/23	10:55	Angela Mook
26/05/23	13:25	Angela Mook
12/06/23	09:15	Angela Mook

Select a meeting:

Confirm Cancel

Back

## Student interface

Student:

User Name:

Password:

↓

**Upcoming meeting:**

Date	Time	Status
22/05/23	10:55	Confirmed

Select above:

★ **Account settings**

**TO DO:**

↓

**CANCEL MEETING**

Date: 22/05/23    Time: 10:55

Reason:

↓

**RESCHEDULE MEETING**

Previous date & time:

New date & time:

View meeting details

Date: 22/05/23 Time: 10:55

Notes:

I want to go over my personal statement...  
....

Go back

Schedule a meeting

**SCHEDULE MEETING**

Date: 12/06/23 Time: 09:15

Notes:

Regarding my teacher recommendations...

done

View all meetings

**ALL MEETINGS**

Date	Time	Status
22/05/23	10:55	Confirmed
26/05/23	13:25	Confirmed
12/06/23	09:15	Pending

Select above:

Cancel Reschedule

View meeting details

Back

★ Account settings

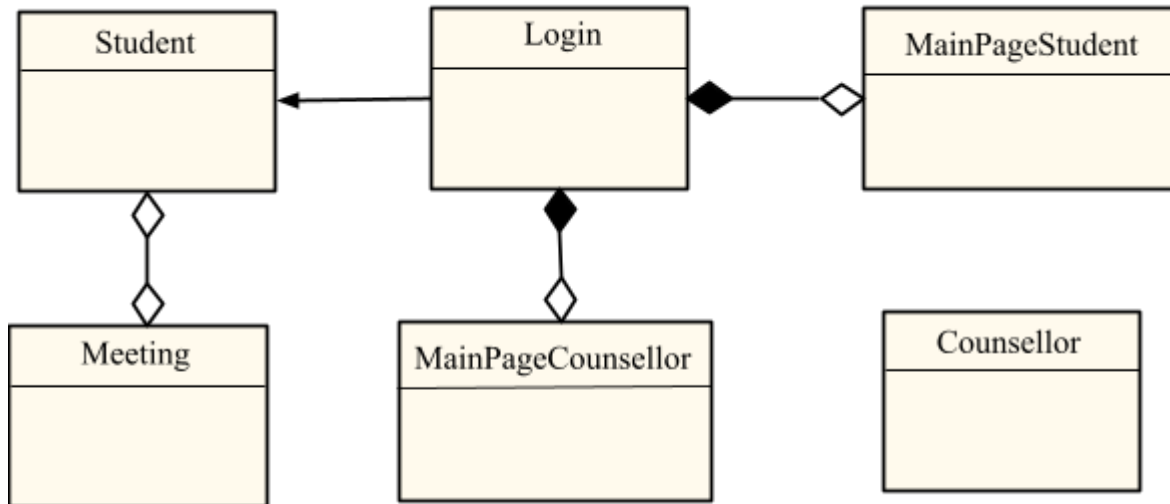
Name: Angela Mook

Universities:

- King's College
- Imperial
- ...

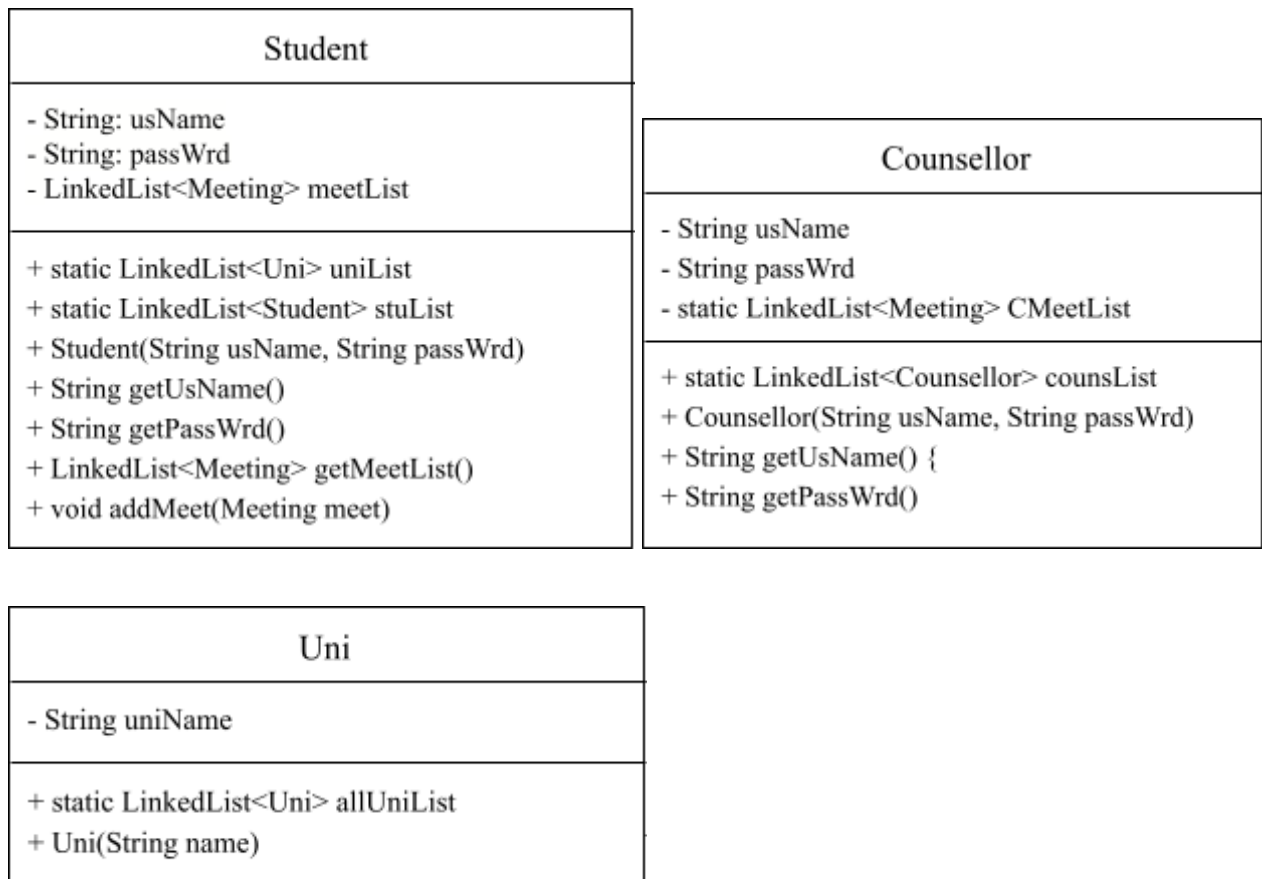
Go back

### Relationships between the classes:



### UML Diagrams of Object Classes:

#### Initial Version



Meeting
<ul style="list-style-type: none"> <li>- Student stu</li> <li>- String notes</li> <li>- String time</li> <li>- String date</li> <li>- String status</li> </ul>
<ul style="list-style-type: none"> <li>+ Meeting(Student stu, String notes, String date, String time)</li> <li>+ String getNotes()</li> <li>+ Student getStu()</li> <li>+ String getTime()</li> <li>+String getDate()</li> <li>+ String getStatus()</li> <li>+ void approve()</li> <li>+ void cancel()</li> </ul>

### Updated Version

While developing my code, I realised:

- Some of the functions of the classes needed some changes
- There was no need for the Uni class anymore

Student
<ul style="list-style-type: none"> <li>- String: usName</li> <li>- String: passWrd</li> <li>+ LinkedList&lt;Meeting&gt; meetList</li> <li>+ LinkedList&lt;String&gt; uniList</li> <li>+ static LinkedList&lt;Student&gt; stuList</li> <li>+ Meeting cancelledMeet</li> </ul>
<ul style="list-style-type: none"> <li>+ Student(String usName, String passWrd)</li> <li>+ String getUsName()</li> <li>+ String getPassWrd()</li> <li>+ void addMeet(Meeting meet)</li> <li>+ static void saveToFile()</li> <li>+ static void loadFromFile()</li> </ul>

Counsellor
<ul style="list-style-type: none"> <li>- String usName = "Ms. Perky"</li> <li>- String passWrd = "Padua High"</li> </ul>
<ul style="list-style-type: none"> <li>+ Counsellor(String usName, String passWrd)</li> <li>+ String getUsName()</li> <li>+String getPassWrd()</li> </ul>



<div>Meeting</div> <div><div><div>- Student stu</div><div>- String notes</div><div>- String time</div><div>- String date</div><div>- String status</div><div>+ static LinkedList&lt;Meeting&gt; allMeet</div></div><div><div>+ Meeting(Student stu, String date, String time, String notes)</div><div>+ String getNotes()</div><div>+ Student getStu()</div><div>+ String getTime()</div><div>+ String getDate()</div><div>+ void setNotes(String notes)</div><div>+ int compareTo(Meeting that)</div><div>+ static void saveToFile()</div><div>+ static void loadFromFile()</div></div></div>	<div>Login</div> <div><div><div>+ String status</div><div>+ static Student stuCurrent</div><div>+ MainPageCounsellor mainPageC</div><div>+ MainPageStudent mainPageS</div></div><div><div>+ Login()</div><div>- void jButton2ActionPerformed()</div><div>- void jButton1ActionPerformed()</div></div></div>
<div>MainPageStudent</div> <div><div><div>- Login login</div></div><div><div><div>+ MainPageStudent(Login login)</div><div>- void renderMeetingList()</div><div>- void renderMeetingList2()</div><div>- void clearMeetingList()</div><div>+ void setVisible(boolean b)</div><div>- void jButton1ActionPerformed()</div><div>- void jButton6ActionPerformed()</div><div>- void jButton8ActionPerformed()</div><div>- void jButton9ActionPerformed()</div><div>- void jButton10ActionPerformed()</div><div>- void jButton11ActionPerformed()</div><div>- void jButton12ActionPerformed()</div><div>- void jButton4ActionPerformed()</div><div>- void jButton5ActionPerformed()</div><div>- void jButton13ActionPerformed()</div><div>- void jButton14ActionPerformed()</div><div>- void jButton15ActionPerformed()</div></div></div></div>	<div>MainPageCounsellor</div> <div><div><div>- Login login</div></div><div><div><div>+ MainPageCounsellor(Login login)</div><div>- void renderMeetingList()</div><div>- void clearMeetingList()</div><div>+ void setVisible(boolean b)</div><div>- void jButton1ActionPerformed()</div><div>- void jButton5ActionPerformed()</div><div>- void jButton6ActionPerformed()</div><div>- void jButton10ActionPerformed()</div><div>- void jButton8ActionPerformed()</div><div>- void jButton4ActionPerformed()</div><div>- void jButton11ActionPerformed()</div><div>- void jButton12ActionPerformed()</div></div></div></div>

### Pseudocode:

STATUS = get item in ComboBox

if STATUS = "Counsellor"

    if TEXTFIELD1 = "Ms. Perky" AND TEXTFIELD2 = "Padua High" then

        show MAINPAGEC

        hide THIS

    end if

end if

if STATUS = "Student"

    loop I from 0 to STULIST.length

        if USNAME = TEXTFIELD1 AND PASSWRD = TEXTFIELD2 then

            STUCURRENT = STULIST[I]

            show MAINPAGES

            hide THIS

        end if

    end loop

end if

→ In the Login class

→ This function allows the user to log in by

    ◆ If they are the counsellor, it checks against the set values

    ◆ If they are a student it loops through the list of all students and checks if the user inputs are the same as any student in the list

INDEX = -1

CURRENT = null

loop for I from 0 to STULIST.length

    if COMBOBOX = STULIST[I].name

        CURRENT = STULIST[I]

    end if

end loop

loop for I from 0 to CURRENT.MEETLIST.length

    TEXTAREA = TEXTAREA +CURRENT. MEETLIST[I].date + " - " +

CURRENT.MEETLIST[I].time

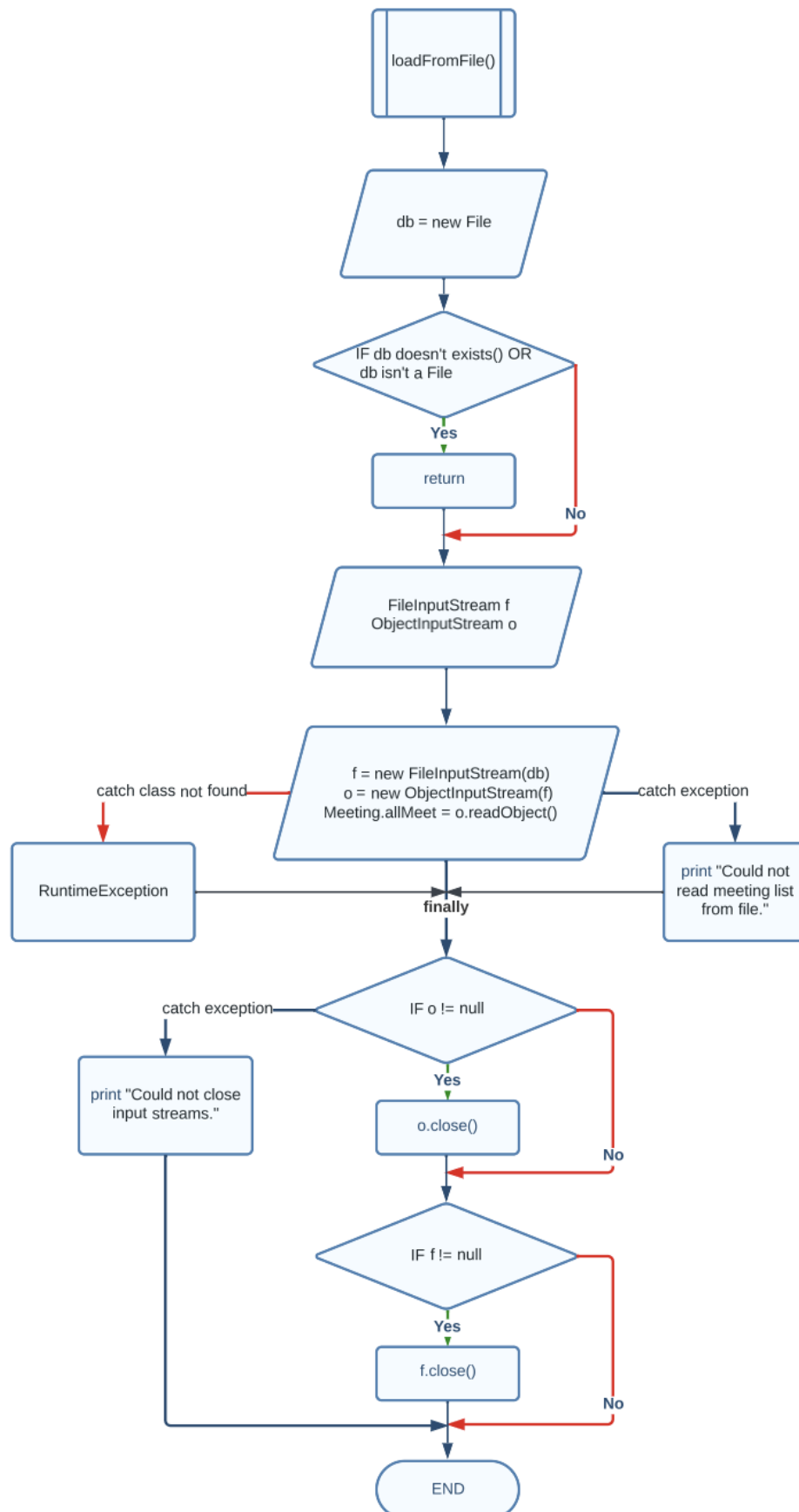
end loop

→ In the MainPageCounsellor class

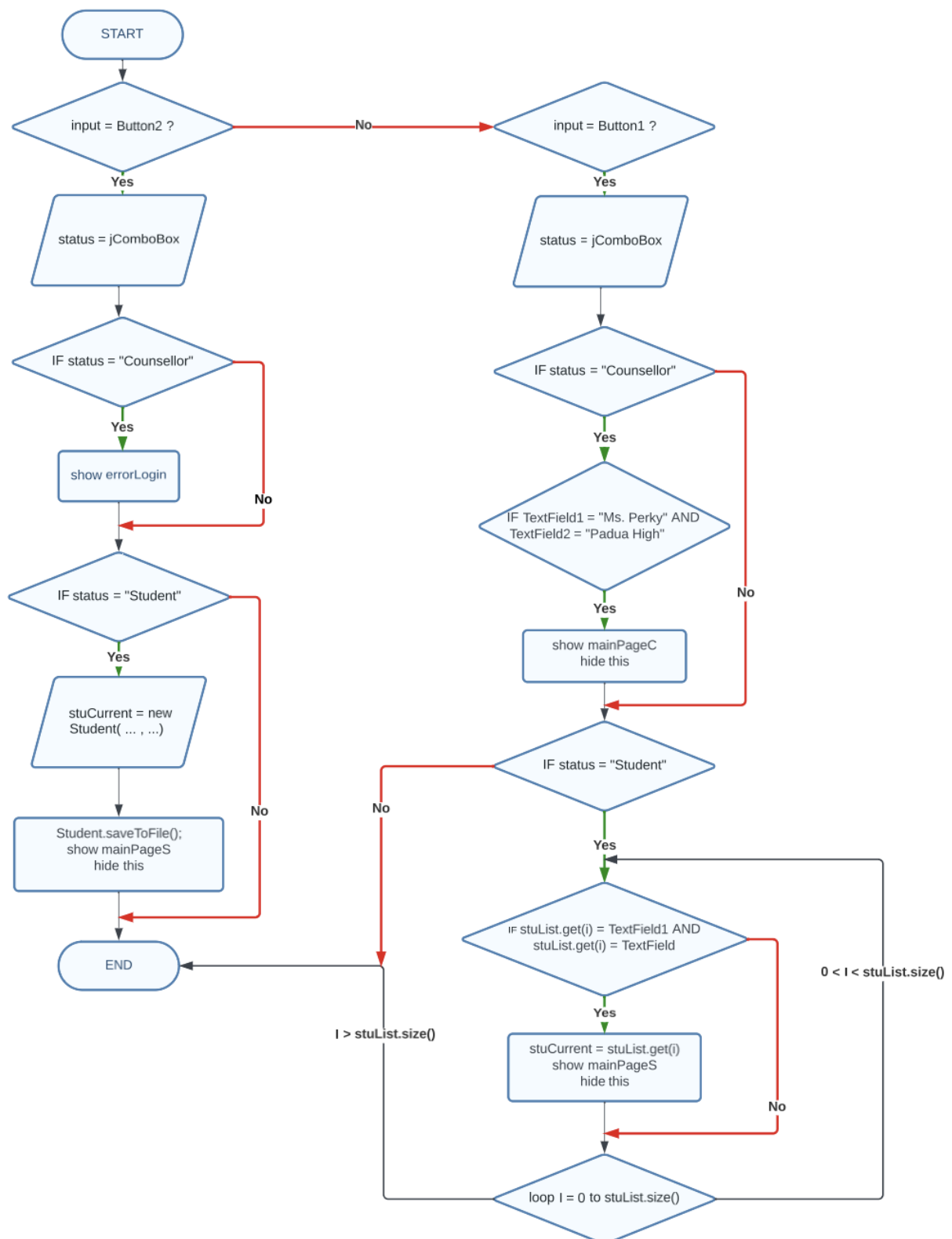
→ This function is executed when the Counsellor searches for meetings with a specific student

## Flowcharts:

→ loadFromFile() technique



→ Beginning of program (login)



### Techniques used:

- If-statements
- Complex if-statements
- Loops
- Nested loops
- Custom methods with parameters and returns
- Sorting
- Searching
- File reading/writing
- Using flags
- Polymorphism
- Encapsulation
- Parsing text in a file or data stream
- Linked list with all methods
- Method overriding
- Association
- Aggregation
- Composition
- Error catching

Word count: 289